# บทที่ 2

# ทฤษฎีการรู้จำตัวพิมพ์อักขระ

## 2.1 ทฤษฎีเบื้องต้น

### 2.1.1 ความหมายของการรู้จำอักขระ

โดยธรรมชาติของมนุษย์เราแล้ว เมื่อได้พบเห็นหรือได้รับฟังสิ่งใดก็ตาม ก็จะเรียนรู้และจดจำสิ่งนั้นไว้ (cognition) จากนั้นเมื่อได้พบเห็นหรือได้รับฟังสิ่งนั้นอีกครั้งหนึ่งก็สามารถที่จะรู้ว่าสิ่งนั้นคืออะไรได้ (recognition) ดังนั้น "การรู้จำอักขระ (character recognition)" หมายถึง การจำลองความสามารถของมนุษย์ดังกล่าว โดยเน้นเฉพาะในเรื่องที่เกี่ยวกับการมองเห็นมาให้เครื่องคอมพิวเตอร์ ให้มีความสามารถในการอ่านและรู้จำอักขระได้

### 2.1.2 ประเภทของการรู้จำอักขระ แบ่งเป็น 2 ประเภท คือ

#### 2.1.2.1 วิธีการรู้จำโดยระบบออนไลน์ (on-line recognition method)

อักขระจะถูกรู้จำได้ในทันที หลังจากขบวนการในการรับอักขระเข้าสู่เครื่องคอมพิวเตอร์เสร็จสิ้นลง อุปกรณ์รับข้อมูลเข้าสู่เครื่องคอมพิวเตอร์สำหรับระบบออนไลน์นี้ ได้แก่ ดิจิไตเซอร์ (digitizer) โดยลักษณะข้อมูลที่เข้าสู่เครื่องคอมพิวเตอร์จะเป็นค่าพิกัดของจุดที่เกิดจากการลากปากกาบนแผ่นดิจิไตเซอร์ และอักขระจะถูกรู้จำได้ในทันทีที่การเขียนอักขระตัวนั้นเสร็จสิ้นลง

#### 2.1.2.2 วิธีการรู้จำโดยระบบออฟไลน์ (off-line recognition method)

อุปกรณ์รับข้อมูลเข้าสู่เครื่องคอมพิวเตอร์สำหรับระบบนี้ ได้แก่ เครื่องกวาดตรวจด้วยแสง (optical scanner) โดยลักษณะข้อมูลที่เข้าสู่เครื่องคอมพิวเตอร์หลังจากผ่านการกวาดตรวจด้วยแสงแล้ว จะอยู่ในรูปที่เป็นภาพบิตเมตริกซ์ของอักขระแต่ละตัวเรียงต่อกันอยู่ สำหรับการรู้จำในระบบออฟไลน์นั้น อักขระที่ถูกเขียนหรือพิมพ์จะยังไม่ถูกรู้จำใน

ทันที คือหลังจากที่อักขระถูกป้อนเข้าสู่เครื่องคอมพิวเตอร์โดยผ่านเครื่องกวาดตรวจด้วยแสงแล้ว ก็จะนำข้อมูลอักขระที่ได้นั้น ไปผ่านขบวนการในการตัดอักขระออกจากประโยคทีละตัวเสียก่อน จากนั้นจึงจะนำเมตริกซ์ของอักขระแต่ละตัวไปประมวลผลเพื่อการรู้จำอีกทีหนึ่ง อักขระที่นำมาประยุกต์ใช้กับการรู้จำในระบบนี้ มี 2 ประเภท คือ

ก. อักขระตัวพิมพ์ (Printed characters)

รูปแบบของอักขระชนิดนี้จะได้มาจากการพิมพ์ต่างๆ เช่น จากเครื่องพิมพ์ดีด หรือ อุปกรณ์การพิมพ์อื่นๆ เป็นต้น โดยมีรูปแบบอักขระที่แน่นอนสำหรับอุปกรณ์การพิมพ์แต่ละชนิด และจากการพิมพ์เหล่านี้จะได้เป็นสิ่งตีพิมพ์ต่างๆ เช่น หนังสือ บทความ หรือใบสำคัญต่างๆ เป็นต้น

การรู้จำอักขระตัวพิมพ์แบ่งออกได้เป็น 2 ประเภท คือ

ก.1 การรู้จำอักขระตัวพิมพ์รูปแบบเดียว (Single-font)

การรู้จำอักขระประเภทนี้มีจุดมุ่งหมายเพื่อให้เครื่องคอมพิวเตอร์สามารถอ่านและรู้จำตัวพิมพ์อักขระได้เฉพาะ รูปแบบของอักขระที่เก็บไว้เพียงรูปแบบเดียวเท่านั้น

ก.2 การรู้จำอักขระตัวพิมพ์หลายรูปแบบ (Multiple-font)

การรู้จำอักขระประเภทนี้มีจุดมุ่งหมายเพื่อให้เครื่องคอมพิวเตอร์สามารถอ่านและรู้จำตัวพิมพ์อักขระได้มากกว่า 1 รูปแบบของอักขระ

ข. อักขระลายมือเขียน (Hand-printed characters)

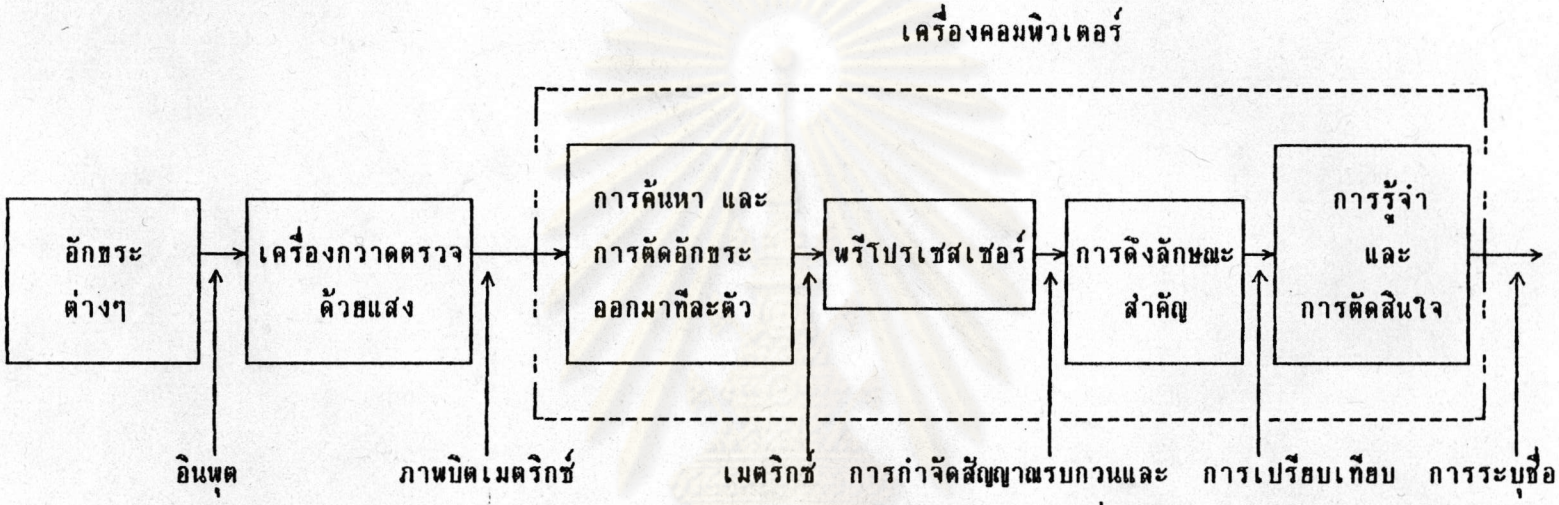
รูปแบบอักขระจะได้จากการเขียนทั่วไป ซึ่งไม่สามารถกำหนดรูปแบบที่แน่นอนได้ ขึ้นกับการเขียนของแต่ละคน (วัลนพ ตันฤดี, 2533)

### 2.1.3 หลักการทำงานของการรู้จำอักขระด้วยแสง

กระบวนการรู้จำอักขระด้วยแสง (optical character recognition) โดยเครื่องคอมพิวเตอร์มีหลักการทำงานดังแสดงในรูปที่ 2.1 ในขั้นแรกอักขระต่างๆ ที่อยู่บนสิ่งตีพิมพ์จะถูกอ่านเข้ามาโดยใช้เครื่องกวาดตรวจด้วยแสงเป็นอุปกรณ์รับข้อมูล ข้อมูลที่เข้าสู่เครื่องคอมพิวเตอร์จะเข้ามาทีละหน้า และอยู่ในรูปที่เป็นภาพบิตเมตริกซ์ของอักขระต่างๆ ที่อยู่บนสิ่งตีพิมพ์นั้น จากนั้นจะมีการค้นหาอักขระและตัดอักขระออกมาทีละตัวจากประโยค แล้วจึงเก็บอักขระที่ตัดออกมาลงเมตริกซ์

เมตริกซ์ของอักขระที่ได้จะผ่านเข้าสู่พรีโปรเซสเซอร์ (preprocessor) เพื่อทำเมตริกซ์ให้มีลักษณะคมชัดขึ้น โดยการกำจัดสัญญาณรบกวน (noise) สัญญาณรบกวนนี้อาจเกิดจากการทำงานของเครื่องกวาดตรวจด้วยแสงในขณะที่กำลังรับข้อมูลเข้า หรืออาจเกิดจากอักขระบนสิ่งตีพิมพ์ที่รับเข้ามามีลักษณะของเส้นไม่เรียบหรืออาจเกิดจากรอยสกปรกบนสิ่งตีพิมพ์ก็ได้ ทำให้เกิดลักษณะการนูนเว้าเล็กๆ บนเส้นแสดงขอบของอักขระ ซึ่งสัญญาณรบกวนนี้จะเป็นผลให้เกิดความผิดพลาดในการรู้จำอักขระขึ้นได้ ดังนั้นจึงจำเป็นจะต้องมีการกำจัดสัญญาณรบกวนออกก่อนที่จะมีการดึงลักษณะสำคัญของอักขระออกมา ตัวอย่างส่วนของเส้นที่มีสัญญาณรบกวนเกิดขึ้นแสดงได้ดังในรูปที่ 2.2(ก) ซึ่งจะเห็นว่ามีลักษณะการนูนเว้าเกิดขึ้น แต่เมื่อกำจัดสัญญาณรบกวนออกแล้ว ลักษณะการนูนเว้าจะหายไปและเปลี่ยนเป็นเส้นตรงดังรูปที่ 2.2(ข) เมื่อกำจัดสัญญาณรบกวนออกจากเมตริกซ์ของอักขระเรียบร้อยแล้ว ก็จะต้องมีการปรับเมตริกซ์นั้นให้มีลักษณะตามที่กำหนด (normalization) ซึ่งมีทั้งการปรับขนาดของอักขระ (size normalization) และการปรับอักขระจากตัวเอียงให้เป็นตัวตรง (skew normalization) เพื่อขจัดปัญหาในกรณีที่อักขระที่รับเข้ามาอาจมีทั้งขนาดเล็กและขนาดใหญ่ หรืออาจมีอักขระที่มีลักษณะเป็นตัวเอียงอยู่ด้วยก็ได้ ซึ่งขนาดและลักษณะการเอียงของอักขระที่แตกต่างกันนี้ อาจมีผลต่อการดึงลักษณะสำคัญออกจากอักขระในขั้นตอนต่อไปได้ คือ อาจทำให้ได้ค่ารหัสที่แตกต่างกันสำหรับอักขระตัวเดียวกันแต่มีขนาดหรือลักษณะการเอียงที่แตกต่างกัน

จากนั้นก็จะเข้าสู่ขั้นตอนของการดึงลักษณะเด่น หรือ ลักษณะสำคัญของอักขระออกมา ขั้นตอนนี้ถือว่าเป็นขั้นตอนที่มีความสำคัญมาก ซึ่งจะมีผลต่อวิธีการที่ใช้ในการรู้จำและอัตราความถูกต้องในการรู้จำด้วย เนื่องจากลักษณะสำคัญของอักขระนี้จะเป็นสิ่งที่ใช้บอกว่าอักขระที่รับเข้ามานั้นเป็นอักขระใด โดยในขั้นตอนของการสร้างพจนานุกรมของอักขระนั้นจะมีการป้อนชุดข้อมูลอักขระตัวอย่างเข้าสู่เครื่องกวาดตรวจด้วยแสง และเมื่อเครื่องคอมพิวเตอร์ได้รับอักขระตัวอย่างมาแล้วก็จะกำจัดสัญญาณรบกวน ปรับเมตริกซ์ให้มีลักษณะตามที่กำหนด จากนั้นก็จะดึงลักษณะสำคัญของอักขระออกมาและจัดเก็บเข้าไว้เป็นพจนานุกรมของอักขระ ต่อมาเมื่อต้องการให้




รูปที่ 2.1 แผนภาพแสดงหลักการทำงานของการรู้จำอักขระด้วยแสง

เครื่องคอมพิวเตอร์อ่านอักขระก็ทำได้โดยวิธีเดียวกัน เพียงแต่หลังจากที่มีการดึงลักษณะสำคัญของอักขระออกมาแล้ว ก็จะนำลักษณะเหล่านั้นไปเปรียบเทียบกับข้อมูลในพจนานุกรมของอักขระ ถ้าลักษณะสำคัญนั้นใกล้เคียงหรือเหมือนกับลักษณะสำคัญของอักขระต้นแบบใดในพจนานุกรม ก็จะถือว่า อักขระต้นแบบนั้นเป็นอักขระที่รู้จำได้ (Suen, Berthod, and Mori, 1980)

(ก) ส่วนของเส้นที่มีสัญญาณรบกวน (ข) ส่วนของเส้นหลังจากกำจัดสัญญาณรบกวน

รูปที่ 2.2 ตัวอย่างการกำจัดสัญญาณรบกวน

2.1.4 <u>เทคนิคการรู้จำอักขระ</u> แบ่งเป็น 2 ประเภท คือ (Suen 1982, quoted in Hor 1985)

2.1.4.1 <u>ประเภทที่นำลักษณะสำคัญทางโกลบอล (global features) มาใช้</u>

เทคนิคประเภทนี้จะพิจารณาลักษณะสำคัญของอักขระในลักษณะของภาพรวม คือ จะมองภาพบิตเมตริกซ์ของอักขระแต่ละตัวในลักษณะเป็นภาพรวม โดยใช้เทคนิคต่อไปนี้ (Suen et al., 1980)

ก. การจับคู่เปรียบเทียบกับอักขระต้นแบบ (template matching and correlations)

เทคนิคนี้จะนำลักษณะของทุกจุดที่อยู่ภายในกรอบ (frame) มาพิจารณา และ สามารถที่จะวัดความคล้ายระหว่างอักขระที่รับเข้ามากับอักขระต้นแบบที่เก็บไว้ได้อย่างง่ายดาย โดยการจับคู่ และเปรียบเทียบจุดที่อยู่ภายในกรอบนั้น

ข. การแปลงและการขยายอนุกรม (transformations and series expansions)

ปัญหาหนึ่งที่เกิดจากการใช้จุด ในลักษณะเช่นที่ใช้กับวิธีการจับคู่เปรียบเทียบกับอักขระต้นแบบ คือขนาดของเวกเตอร์ที่แสดงลักษณะมีขนาดใหญ่ เนื่องจากต้องเก็บทุกจุดภายในกรอบ ดังนั้นจึงได้นำเทคนิคนี้มาใช้เพื่อช่วยลดขนาดของเวกเตอร์ และสามารถใช้เทคนิคนี้ในการดึงลักษณะสำคัญของภาพออกมาได้ ไม่ว่าลักษณะของภาพจะเปลี่ยนไปอย่างไร เช่น เกิดการเอียง หรือการหมุนของภาพ ตลอดจนขนาดของภาพที่เปลี่ยนไปด้วย

จากนั้นการจะตัดสินใจว่า รูปแบบที่ไม่ทราบกับรูปแบบต้นแบบมีความคล้ายกันเพียงไรทำโดยการนำทฤษฎีทางสถิติมาประยุกต์ใช้

วิธีการรู้จำอักขระที่ใช้สารสนเทศรวมของรูปแบบนั้น ง่ายในการหาลักษณะสำคัญของอักขระ และมีผลกระทบน้อยจากการรบกวนของลักษณะในส่วนต่างๆ ของภาพ แต่ถ้ามีการสูญเสียลักษณะสำคัญในส่วนต่างๆ ของภาพไปแล้ว ก็จะมีผลกระทบอย่างมากต่อความถูกต้องในการรู้จำ (วัลนพ ตันฤดี, 2533)

2.1.4.2 <u>ประเภทที่นำลักษณะสำคัญทางโลคอล (local features) มาใช้</u>

เทคนิคประเภทนี้อาจเรียกว่า การวิเคราะห์โครงสร้างรูปแบบของอักขระ (Structural analysis) หรือ การวิเคราะห์แบบซินแต็กติก (Syntactic analysis) โดยจะนำคุณสมบัติของแต่ละส่วนย่อยที่ประกอบเป็นอักขระมาใช้ เช่น ทิศทางการลากเส้นแสดงอักขระ ความโค้งเว้า จุดปลาย และ การตัดกันของเส้น (Kushnir, Abe, and Matsumoto, 1985) เทคนิคนี้กำลังได้รับความสนใจเป็นอย่างมากในปัจจุบัน เนื่องจากมีขีดความสามารถที่ทฤษฎีทางสถิติยังขาดอยู่ คือความสามารถในการจัดการโครงสร้าง ซึ่งจะดูถึงส่วนของเส้น (line segments) เส้นแสดงขอบ (contour lines) และเส้นแสดงโครงร่างที่ได้จากการลดความหนาของรูปแบบ (skeleton lines) นอกจากนี้ยังช่วยลดขนาดของเซตที่แสดงลักษณะได้ด้วย

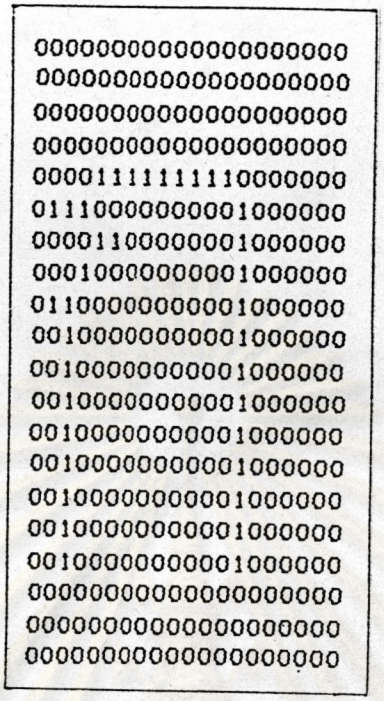
เทคนิคประเภทนี้แบ่งตามลักษณะที่ถูกดึงมาใช้เป็น 2 ประเภท คือ

ก. การวิเคราะห์เส้นแสดงโครงร่างอักขระ (skeleton)

ในการรู้จำอักขระนั้น ปัญหาหนึ่งที่ต้องประสบอยู่เสมอ คือ ขนาดความหนาบางของอักขระที่จะรู้จำมีขนาดไม่เท่ากัน ทำให้รหัสที่ใช้แทนอักขระรูปเดียวกัน แต่มีความหนาบางต่างกันเป็นรหัสที่แตกต่างกันออกไปด้วย ดังนั้นเพื่อขจัดปัญหานี้จึงใช้วิธีการแปลงภาพบิตเมตริกซ์ของอักขระที่จะรู้จำ ให้เหลือแต่โครงร่างที่มีความหนาบางเท่ากันตลอด ไม่ว่าอักขระนั้นจะมีความหนาบางเริ่มแรกเท่าใดก็ตาม การแปลงภาพบิตเมตริกซ์ของอักขระให้เหลือแต่เพียงโครงร่างของอักขระนี้อาจเรียกได้ว่า เป็นการลดความหนาของเมตริกซ์ของอักขระนั่นเอง (thining) ตัวอย่างของเส้นแสดงโครงร่างอักขระแสดงได้ดังรูปที่ 2.3

เมื่อได้เส้นแสดงโครงร่างอักขระแล้ว ก็จะนำเส้นแสดงโครงร่างอักขระนี้ไปใช้ในการวิเคราะห์ โดยลักษณะสำคัญที่อาจนำมาใช้ในการวิเคราะห์ได้แก่ จุดที่เกิดจากการตัดกันของเส้นสองเส้น จุดที่เส้นสามเส้นมาเชื่อมต่อกัน และ จุดปลายของเส้น เป็นต้น นอกจากนี้อาจมีการให้รหัสแก่เส้นแสดงโครงร่างอักขระ เช่น ให้รหัสที่ใช้แสดงทิศทางแก่เส้นแสดงโครงร่างอักขระตามลักษณะทิศทางที่เส้นแสดงโครงร่างอักขระเปลี่ยนแปลงไป หรือให้รหัสที่แสดงถึงความหนาแน่นและการกระจายของจุดที่เป็น 1 ในภาพบิตเมตริกซ์ของอักขระตามแนวแถวและแนวสดมภ์ เป็นต้น นอกจากนี้แล้วอาจมีการนำเส้นแสดงโครงร่างอักขระไปใช้หาขนาดความกว้างและความสูงของอักขระสำหรับนำมาใช้ในการวิเคราะห์ก็ได้

เนื่องจากเทคนิคนี้ เมื่อนำมาใช้กับอักขระซึ่งมีโครงสร้างที่ซับซ้อน และประกอบด้วยเส้นโค้งเป็นส่วนใหญ่ ดังเช่น อักขระไทย แล้ว จะมีผลให้การที่จะได้เส้นแสดงโครงร่างอักขระที่สมบูรณ์เป็นไปได้ลำบาก ตัวอย่างเช่น ในกรณีที่อักขระมีลักษณะคล้ายกัน เช่น ก กับ ถ ซึ่งเมื่อพิจารณาแล้วจะเห็นได้ว่า อักขระทั้งสองมีลักษณะที่แตกต่างกันอยู่เพียงประการเดียวเท่านั้น คือ ตัวหนึ่งมีหัว และอีกตัวหนึ่งไม่มีหัว ส่วนลักษณะอื่นนอกจากนั้นเหมือนกันหมด ซึ่งในกรณีเช่นนี้ถ้าหากว่า ถ มีส่วนที่เป็นหัวเล็กมาก ก็จะมีผลทำให้ได้เส้นแสดงโครงร่างอักขระที่เหมือนกับ ก ได้ ดังนั้นเทคนิคนี้จึงเหมาะที่จะนำไปประยุกต์ใช้กับอักขระประเภทที่ประกอบด้วยเส้นตรงเป็นส่วนใหญ่ เช่น อักขระจีน อักขระเกาหลี เป็นต้น

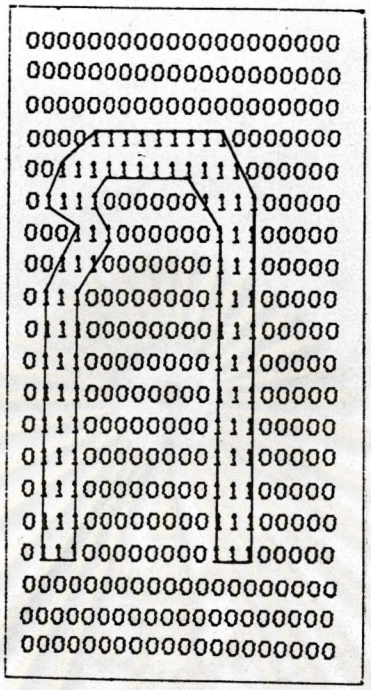



รูปที่ 2.3 ตัวอย่างภาพบิตเมตริกซ์ของเส้นแสดงโครงร่างอักขระ (ชมทิน พรพนมชัย, 2529) ของตัวเลข 1 ที่เรียงติดต่อกัน แสดงให้เห็นเส้นแสดงโครงร่างอักขระ

ข. การวิเคราะห์เส้นแสดงขอบของอักขระ (contour)

เส้นแสดงขอบอักขระ ดังในรูปที่ 2.4 หมายถึง เส้นต่างๆ ที่ประกอบกันขึ้นเป็นอักขระ 1 ตัว เส้นเหล่านี้จะมีลักษณะโครงสร้างความโค้งเว้าที่ซับซ้อนแตกต่างกันขึ้นกับว่าเป็นอักขระใด เช่น อักขระ ก จะเห็นว่ามีลักษณะโครงสร้างความโค้งเว้าไม่ซับซ้อนมากนัก คือมีเพียงไม่กี่ส่วนของอักขระที่มีความโค้งเว้า แต่ถ้าเป็น อักขระ ต จะพบว่าลักษณะโครงสร้างความโค้งเว้ามีความซับซ้อนมากกว่า อักขระ ก คือมีความโค้งเว้าที่หลายส่วนของอักขระ เป็นต้น

เทคนิคนี้ จะนำลักษณะโครงสร้างความโค้งเว้าของอักขระมาใช้เป็นลักษณะสำคัญในการวิเคราะห์ จึงเหมาะที่จะนำไปประยุกต์ใช้กับอักขระประเภทที่ประกอบด้วยเส้นโค้งเป็นส่วนใหญ่ เช่น อักขระไทย เป็นต้น




รูปที่ 2.4 ตัวอย่างภาพบิตเมตริกซ์ของเส้นแสดงขอบของอักขระ (ชมทิพ พรพนมชัย, 2529) ของตัวเลข 1 ที่เรียงติดต่อกันตามแนวเส้น แสดงให้เห็นเส้นแสดงขอบของอักขระ

## 2.2 การวิจัยเกี่ยวกับการรู้จำอักขระไทยในอดีต

### พิพัฒน์ หิรัญย์วณิชชากร และคนอื่น ๆ (1982, 1984, and 1985)

ค.ศ.1982 เสนอวิธีการรู้จำตัวพิมพ์อักขระไทยโดยการวิเคราะห์เส้นแสดงขอบของอักขระ และให้รหัสเพื่อแสดงทิศทางที่เปลี่ยนไปตามลักษณะความโค้งเว้าของเส้นแสดงขอบของอักขระ โดยนำหลักพื้นฐานทางคณิตศาสตร์มาใช้เพื่อกำจัดสัญญาณรบกวน แล้วตัดส่วนความโค้งเว้าของเส้นแสดงขอบของอักขระออกเป็นส่วนย่อย จากนั้นจึงนำลักษณะทางเรขาคณิตของส่วนโค้งที่ตัดได้ ได้แก่ ความยาวของส่วนโค้ง ระยะห่างระหว่างจุดเริ่มต้นกับจุดสิ้นสุดของส่วนโค้ง ระยะห่างระหว่างจุดเริ่มต้นกับจุดศูนย์ถ่วง (centroid) ของส่วนโค้ง ระยะห่างระหว่างจุดสิ้นสุดกับจุดศูนย์ถ่วงของส่วนโค้ง และ มุมที่เกิดจากการทำมุมกันของเส้นที่ลากจากจุดศูนย์ถ่วงไปยังจุดเริ่มต้นกับเส้นที่ลากจากจุดศูนย์ถ่วงไปยังจุดสิ้นสุดของส่วนโค้ง ลักษณะทางเรขาคณิตเหล่านี้จะนำมาใช้ในการคำนวณค่าความคล้าย (similarity) ระหว่างส่วนโค้งของอักขระที่รับเข้ามากับส่วนโค้งของอักขระต้นแบบ เมื่อได้ค่าความคล้ายระหว่างแต่ละคู่ส่วนโค้งของอักขระแล้ว

ก็จะนำไปใช้ในการคำนวณค่าความคล้ายระหว่างอักขระ ผลที่ได้จากการวิจัยมีความถูกต้อง 98.2 เปอร์เซ็นต์

ค.ศ.1984 เสนอวิธีการรู้จำตัวพิมพ์อักขระไทยโดยการวิเคราะห์เส้นแสดงขอบของอักขระ และนำรหัสแบบลูกโซ่ของฟรีแมนกับความแตกต่างของทิศทางของเส้นแสดงขอบของอักขระมาใช้ เพื่อกำจัดสัญญาณรบกวนและตัดส่วนความโค้งเว้าของอักขระ โดยใช้จุดที่เส้นแสดงขอบของอักขระมีการเปลี่ยนทิศทาง ทำให้ได้ส่วนโค้งย่อยออกมา จากนั้นจะดึงลักษณะทางโลคอลของส่วนโค้งเหล่านั้นออกมาได้แก่ความยาวของแต่ละส่วนโค้งย่อย ลักษณะทางโลคอลที่ดึงออกมานี้จะนำมาใช้ในการคำนวณค่าความคล้ายระหว่างส่วนโค้งของอักขระที่รับเข้ามากับส่วนโค้งของอักขระต้นแบบ เมื่อได้ค่าความคล้ายระหว่างแต่ละคู่ส่วนโค้งของอักขระแล้ว ก็จะนำไปใช้ในการคำนวณค่าความคล้ายระหว่างอักขระ ผลที่ได้จากการวิจัยมีความถูกต้อง 99.4 เปอร์เซ็นต์

ค.ศ.1985 เสนอวิธีการรู้จำลายมือเขียนอักขระไทยโดยนำลักษณะทางโลคอลมาใช้และนำรหัสแบบลูกโซ่ของฟรีแมน กับ ความแตกต่างของทิศทางของเส้นแสดงขอบของอักขระมาใช้เพื่อตัดส่วนความโค้งของอักขระ การสร้างพจนานุกรมเริ่มจากการคำนวณค่าความคล้ายระหว่างส่วนโค้งของอักขระที่รับเข้ามากับส่วนโค้งของอักขระต้นแบบ โดยในการคำนวณค่าความคล้ายนี้จะนำลักษณะทางโลคอลของส่วนโค้งมาใช้ ได้แก่ ความยาวของแต่ละส่วนโค้งย่อย จากนั้นจึงนำค่าความคล้ายระหว่างแต่ละคู่ส่วนโค้งของอักขระที่ได้ มาใช้ในการคำนวณค่าความคล้ายระหว่างอักขระ ส่วนโค้งที่คล้ายกันจะได้จากกลุ่มที่ต่างกันของอักขระ เพื่อสร้างพจนานุกรมของส่วนโค้ง และ สร้างพจนานุกรมของอักขระซึ่งประกอบด้วยรายชื่อส่วนโค้งของอักขระ ผลที่ได้จากการวิจัยมีความถูกต้อง 88.9 เปอร์เซ็นต์

<u>ชม กิ้มปาน และคนอื่น ๆ (1983, 1987, and 1989)</u>

ค.ศ.1983 นำวิธีการซ้อนทับ (matching method) มาใช้รู้จำตัวพิมพ์อักขระไทยโดยแบ่งเป็น 2 ขั้นตอน ในขั้นตอนของการแบ่งกลุ่มอักขระ รูปร่างปลีกย่อยที่ไม่จำเป็นและสัญญาณรบกวนจะถูกกำจัด โดยวิธีการทำอักขระให้เบลอ (blurring method) วิธีการทำอักขระให้เบลอนั้นทำโดยการค้นหาข้อมูลที่มีค่าเป็น 1 ในเมตริกซ์ของอักขระทีละแถว โดยเริ่มจากแถวแรกไปจนถึงแถวสุดท้าย ถ้าพบข้อมูลที่มีค่าเป็น 1 ก็จะเปลี่ยนข้อมูลที่อยู่รอบๆ ตำแหน่งนั้นให้มีค่าเป็น 1 ด้วย อักขระที่เบลอแล้ว (blurred characters) จะถูกแบ่งเป็นกลุ่มๆ โดยใช้วิธีวัดค่าสัมประสิทธิ์ความเหมือนของแต่ละรูปแบบ และใช้การกระจายแบบคาร์ฮูเนนโลบหารูปแบบมาตรฐานเพื่อเป็นตัวแทนของแต่ละกลุ่ม ในขั้นตอนของการแยกอักขระออกจากกลุ่มใช้วิธีซ้อนทับ

เป็นส่วนๆ (subpattern matching) เพื่อแยกอักขระออกจากกัน ผลที่ได้จากการวิจัยมีความถูกต้อง 98.2 เปอร์เซ็นต์

ค.ศ.1987 เสนอการรู้จำตัวพิมพ์อักขระไทย โดยใช้การกระจายแบบคาร์ฮูเนนโลบ แบ่งเป็น 2 ขั้นตอน ขั้นตอนแรกเป็นการแบ่งกลุ่มอักขระ ไอเกนเวกเตอร์ที่มีค่าไอเกนสูงสุดในการกระจายแบบคาร์ฮูเนนโลบถูกนำมาใช้เป็นรูปแบบมาตรฐานของแต่ละกลุ่มอักขระ ขั้นตอนที่สองเป็นการแยกอักขระออกจากกลุ่ม ไอเกนเวกเตอร์ที่ไม่ได้ใช้ในขั้นแรกจะถูกนำมาใช้โดยหาฟังก์ชั่นการตัดสินใจแบบเชิงเส้นบนระนาบไอเกนเวกเตอร์ที่ได้จากการกระจายแบบคาร์ฮูเนนโลบ เพื่อแยกอักขระออกจากกลุ่ม ผลที่ได้จากการวิจัยมีความถูกต้อง 98 เปอร์เซ็นต์

ค.ศ.1989 เสนอการรู้จำตัวพิมพ์อักขระไทยโดยใช้วิธีค้นหาจุดเด่นของอักขระ (feature concentrated method) ในการเตรียมข้อมูลของอักขระ จากนั้นจะเป็นการหาลักษณะเด่นของอักขระ โดยการหาคอนเซ็นเทรทเวิร์ด (concentrated word) ณ จุดศูนย์ถ่วง ผลที่ได้จากการวิจัยมีความถูกต้อง 90 เปอร์เซ็นต์

Ding-Chern, Hor (1985)

เสนอระบบการรู้จำลายมือเขียนตัวเลขไทยที่ประกอบด้วยหลายขั้นตอน ซึ่งในขั้นแรกนำวิธีการจับคู่อย่างง่าย (simple similarity matching) มาใช้ เพื่อเลือกรูปแบบที่น่าจะเป็นไปได้ที่ดีที่สุดเก็บไว้ 9 รูปแบบ และนำเทคนิคที่ใช้ลักษณะทางโกลบอล เพื่อดึงลักษณะสำคัญมาใช้สำหรับการรู้จำในขั้นแรก ต่อมาจึงได้นำการวิเคราะห์โครงสร้างแบบซินแท็คทิคทรี (syntactic tree) ซึ่งได้รวมเอา อัลกอริทึมการติดตามรหัสแบบลูกโซ่ของฟรีแมน (Freeman chain code tracing algorithm) การดึงพริมิทีฟ (primitive) การทำโพสฟิคทรี (postfix tree) การคำนวณระยะห่างระหว่างทรี (tree) เข้าไว้ด้วยสำหรับการรู้จำในขั้นสุดท้ายนี้ ผลที่ได้จากการวิจัยมีความถูกต้อง 99.5 เปอร์เซ็นต์

ชมทิน พรพนมชัย (1986)

เสนอวิธีการรู้จำอักขระไทยซึ่งอยู่ในรูปแบบของตัวพิมพ์ดีด โดยอักขระที่นำมาใช้ได้มาจากการจัดเตรียมให้อยู่ในรูปของภาพบิตเมตริกซ์ และ เป็นอักขระที่ปราศจากสัญญาณรบกวน ในเบื้องต้นได้สร้างอักขระภาษาไทยในรูปตัวพิมพ์จำนวน 5 ชุด สำหรับใช้เป็นอักขระต้นแบบในการหาผลลัพธ์ของการรู้จำ วิธีที่ใช้ในการรู้จำอักขระแบ่งเป็น 3 ขั้นตอน คือ ในขั้นแรกจะ

เป็นการเปลี่ยนภาพบิตเมตริกซ์ของอักขระให้เป็นเส้นแสดงโครงร่างอักขระ ซึ่งถือได้ว่าเป็นขั้นตอนที่ใช้ในการลดความหนาของอักขระ ในขั้นที่สองเป็นการเปลี่ยนเส้นแสดงโครงร่างอักขระให้อยู่ในรูปของรหัส ซึ่งรหัสเหล่านี้ หมายถึง ลักษณะของอักขระตามแนวแถวและแนวสดมภ์ ส่วนในขั้นสุดท้ายจะเป็นขั้นตอนของการสร้างความสัมพันธ์ระหว่างรหัสในขั้นที่สองกับอักขระต้นแบบที่สร้างไว้ ผลที่ได้จากการวิจัยมีความถูกต้อง 70 เปอร์เซ็นต์

วัลนพ ตันฤดี (1990)

เสนอระบบการรู้จำลายมือเขียนอักขระไทยที่ประกอบด้วยหลายขั้นตอน ซึ่งในขั้นตอนแรกเป็นการแบ่งกลุ่มรูปแบบคร่าวๆ ตามตำแหน่งจุดปลายทั้งสองเป็น 3 กลุ่มย่อย โดยกำหนดขอบเขตระดับการเขียนไว้ล่วงหน้า ต่อมาเป็นการหาลักษณะเด่นของรูปแบบตามรหัสแบบลูกโซ่ของฟรีแมน เพื่อกำหนดการแบ่งกลุ่มเป็น 32 กลุ่มอักขระ ขั้นตอนการรู้จำรูปแบบไดนามิกโปรแกรมมิงระหว่างรูปแบบในกลุ่มทำเพื่อหารูปแบบอ้างอิงที่มีความแตกต่างน้อยที่สุด ผลที่ได้จะผ่านการตรวจสอบอีกครั้งหนึ่งจากภาคจัดการผลลัพธ์ระหว่างรูปแบบในกลุ่มใกล้เคียง ผลที่ได้จากการวิจัยมีความถูกต้อง 98.5 เปอร์เซ็นต์

## 2.3 เทคนิคการวิเคราะห์เส้นแสดงขอบของอักขระ

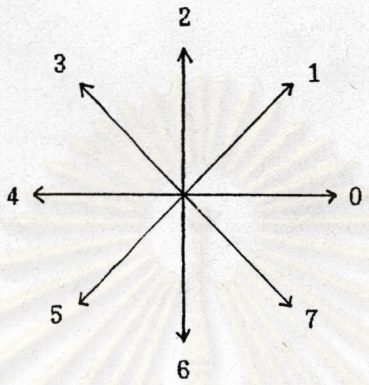
เป็นเทคนิคประเภทที่นำลักษณะทางโลคอลมาใช้ คือจะมีการดึงลักษณะสำคัญของอักขระออกมา และเนื่องจากอักขระไทยส่วนใหญ่ประกอบขึ้นมาจากเส้นโค้ง ซึ่งมีโครงสร้างที่ซับซ้อนโดยลักษณะเฉพาะของเส้นโค้ง เป็นสิ่งที่มีความหมายสำคัญต่อการจัดแบ่งกลุ่มส่วนโค้ง ที่มีลักษณะคล้ายกัน และต่อการสร้างพจนานุกรมต้นแบบด้วย

จากการพิจารณางานวิจัยต่างๆ เกี่ยวกับการรู้จำอักขระไทยนั้นพบว่า เทคนิคการวิเคราะห์เส้นแสดงขอบของอักขระ เป็นเทคนิคหนึ่งที่นับได้ว่ามีประสิทธิภาพสำหรับนำมาใช้ในการรู้จำ (Hiranvanichakorn, P., Agui, T., and Nakajima, M., 1984, 1985) โดยในงานวิจัยทั้งสองชิ้นนั้น ได้นำรหัสแบบลูกโซ่ของฟรีแมนกับความแตกต่างของทิศทางของเส้นแสดงขอบของอักขระ มาใช้ในการตัดส่วนความโค้งเว้าของเส้นแสดงขอบของอักขระออกเป็นส่วนโค้งย่อย จากนั้นก็จะมีการดึงลักษณะสำคัญของส่วนโค้งย่อยเหล่านั้นออกมาได้แก่ ความยาวของแต่ละส่วนโค้งย่อย ลักษณะสำคัญนี้จะถูกนำไปใช้ในการคำนวณค่าความคล้ายระหว่างส่วนโค้งและระหว่างอักขระ เนื่องจากวิธีในการวิเคราะห์ลักษณะสำคัญที่ได้นำมาใช้ในงานวิจัยทั้งสองนั้นเป็นวิธีที่ค่อนข้างจะซับซ้อนและใช้เวลาในการประมวลผลค่อนข้างมาก แต่อย่างไรก็ตามก็ยังเป็นวิธีที่ทำให้เกิดประสิทธิภาพในการนำไปใช้เพื่อการรู้จำอักขระไทย ดังนั้นในงานวิจัยนี้จึงได้มีการปรับปรุงวิธีในการวิเคราะห์ลักษณะสำคัญที่ใช้กับงานวิจัยทั้งสองนั้นเพื่อให้มีความซับซ้อนลดน้อยลง และเหมาะสำหรับการนำไปใช้เพื่อการรู้จำตัวพิมพ์อักขระไทยหลายรูปแบบ

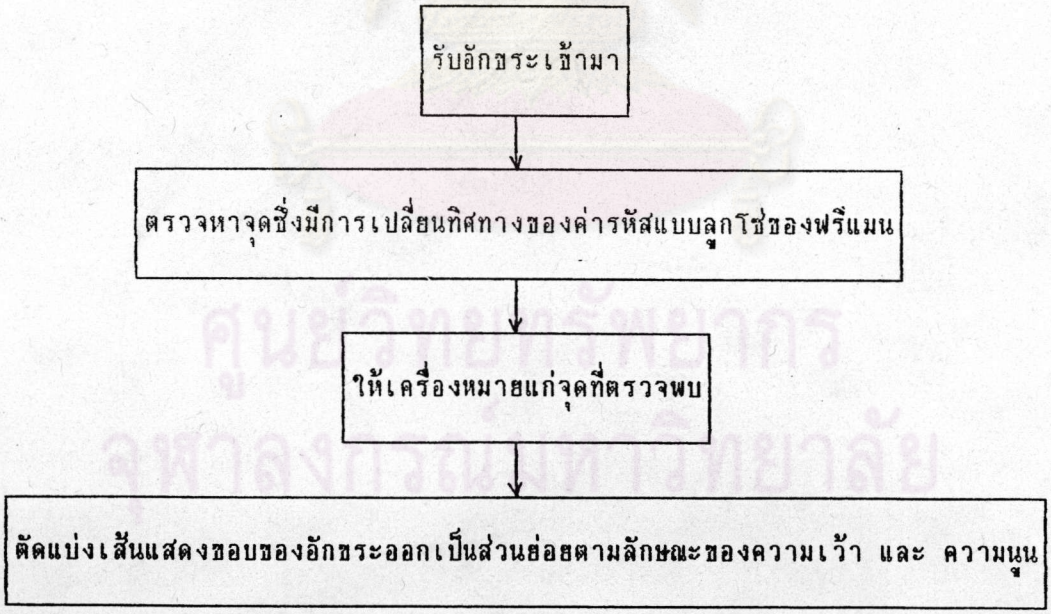
ในการวิจัยนี้ สิ่งที่นำมาใช้เพื่อตัดแบ่งเส้นแสดงขอบของอักขระ ซึ่งมีลักษณะทั้งที่เป็นส่วนเว้า (concavity) และ ส่วนนูน (convexity) ออกเป็นส่วนย่อย คือ รหัสแบบลูกโซ่ของฟรีแมน ซึ่งเป็นรหัสทิศทางที่มีทิศทางทั้งหมด 8 ทิศทาง ดังในรูปที่ 2.5 โดยจะกำหนดรหัสนี้ให้กับทุกจุดบนเส้นแสดงขอบของอักขระ จากนั้นก็จะตรวจหาจุดที่ทำให้เกิดการเปลี่ยนแปลงทิศทางบนเส้นแสดงขอบของอักขระโดยพิจารณาจากค่ารหัสทิศทาง $F_i$ ที่เปลี่ยนไป เมื่อได้จุดที่ทำให้เกิดการเปลี่ยนแปลงทิศทางบนเส้นแสดงขอบของอักขระแล้ว ก็จะมีการกำหนดค่าพร้อมทั้งเครื่องหมาย + และ เครื่องหมาย - ให้กับจุดเปลี่ยนทิศทางเหล่านั้น เพื่อบอกถึงลักษณะการเปลี่ยนทิศทางว่ามีลักษณะตามเข็มนาฬิกาหรือทวนเข็มนาฬิกา โดยเครื่องหมายลบ (-) จะถูกกำหนดให้กับจุดที่ทำให้เกิดการเปลี่ยนทิศตามเข็มนาฬิกา และเครื่องหมายบวก (+) จะถูกกำหนดให้กับจุดที่ทำให้เกิดการเปลี่ยนทิศทวนเข็มนาฬิกา จากนั้นก็จะเป็นขั้นตอนของการนำค่ารหัสทิศทาง $F_i$ เครื่องหมาย + และ เครื่องหมาย - มาใช้ในการกำหนดจุดบ่งความนูน (เครื่องหมาย +) และ จุดบ่งความเว้า (เครื่องหมาย -) ซึ่งจุดบ่งความนูนและจุดบ่งความเว้าบนเส้นแสดงขอบของอักขระที่ได้ในขั้นสุดท้ายนี้ จะถูกนำมาใช้ประโยชน์ในการตัดแบ่งเส้นแสดง

ขอบของอักขระออกเป็นส่วนย่อยตามลักษณะของความเว้า (concave arc) และความนูน (convex arc) ต่อไป

จากขั้นตอนที่กล่าวมาทั้งหมดนี้สามารถสรุปออกมาเป็นแผนภาพได้ ดังแสดงในรูปที่ 2.6



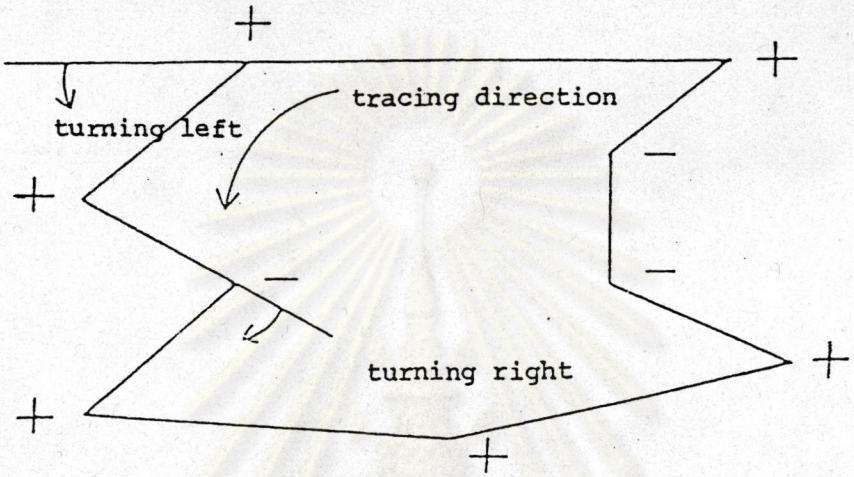

รูปที่ 2.5 รหัสแบบลูกโซ่ของฟรีแมน ($F_i$)



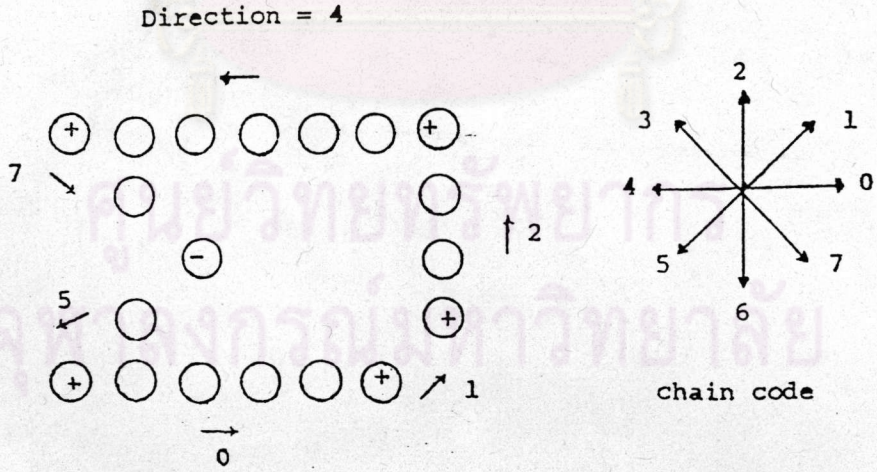

รูปที่ 2.6 แผนภาพแสดงการตัดแบ่งเส้นแสดงขอบของอักขระออกเป็นส่วนย่อยตามลักษณะของความเว้า และ ความนูน (Hiranvanichakorn et al., 1984)

### 2.3.1 การตัดแบ่งเส้นแสดงขอบของอักขระออกเป็นส่วนโค้งย่อย

ดังในรูปที่ 2.7 จะเห็นว่าส่วนของรูปที่มีลักษณะของความเว้า และ ความนูน สามารถจะถูกตัดแบ่งออกเป็นส่วนย่อยได้โดยการให้เครื่องหมาย - หรือ เครื่องหมาย + ณ จุดที่มีการเปลี่ยนแปลงทิศทางว่าจะตามเข็มหรือทวนเข็มนาฬิกา ตามลำดับ




รูปที่ 2.7 การให้เครื่องหมายแก่จุดที่มีการเปลี่ยนแปลงทิศทาง




(ก) (ข)

รูปที่ 2.8 (ก) การให้เครื่องหมายตามการเปลี่ยนทิศทางของเส้นแสดงขอบของรูปที่ถูกเข้ารหัสทิศทางตามรูป (ข)

(ข) รหัสทิศทางแบบลูกโซ่ของฟรีแมน

แนวความคิดข้างต้นนี้สามารถประยุกต์ใช้กับเส้นแสดงขอบของรูปในเชิงดิจิตัล (digital contours) ได้ เพราะลำดับของจุดซึ่งประกอบกันขึ้นเป็นเส้นแสดงขอบของรูป (contour pixels) ที่มีค่ารหัสแบบลูกโซ่ของฟรีแมนเหมือนกัน สามารถทำให้เกิดเป็นเส้นตรงขึ้น 1 เส้นได้ และทิศทางของเส้นตรงนี้ก็จะทราบได้จากค่ารหัสแบบลูกโซ่ของฟรีแมนนั่นเอง ดังนั้นถ้าหากให้เครื่องหมาย - และ เครื่องหมาย + แก่จุดที่ประกอบกันขึ้นเป็นเส้นแสดงขอบของรูป ณ ตำแหน่งที่มีการเปลี่ยนแปลงทิศทาง (ค่ารหัสแบบลูกโซ่ของฟรีแมนเปลี่ยน) ดังแสดงในรูปที่ 2.8(ก) ก็จะทำให้สามารถตัดแบ่งเส้นแสดงขอบของรูปในเชิงดิจิตัลออกเป็นส่วนย่อยได้ตามลักษณะของความเว้า และ ความนูน

กำหนด $P_n = (P_{xn}, P_{yn}) \quad : n = 1, \ldots, N$

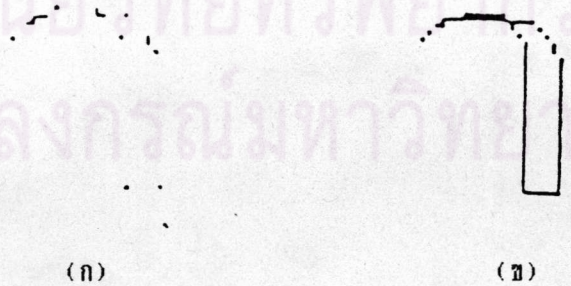
โดย $P_n$ เป็นจุดที่ n บนเส้นแสดงขอบของอักขระบนระนาบ X-Y

$N$ เป็นจำนวนจุดทั้งหมดบนเส้นแสดงขอบของอักขระ

และ $DD_{i,j}$ เป็นระยะทางระหว่างจุดที่ i และ จุดที่ j

เมื่อ $DD_{i,j}$ = (จำนวนจุดซึ่งอยู่ระหว่างจุดที่ i และ จุดที่ j) + 1

จุดซึ่งมีการเปลี่ยนแปลงทิศทางตามรหัสแบบลูกโซ่ของฟรีแมน และไม่ใช่จุดซึ่งเป็นสัญญาณรบกวนจะถูกตรวจพบได้ ในรูปที่ 2.9(ก) จะแสดงให้เห็นถึงจุดที่มีการเปลี่ยนแปลงทิศทางบนเส้นแสดงขอบของอักขระที่ถูกตรวจพบได้ จากอักขระในรูปที่ 2.9(ข)



(ก) (ข)

รูปที่ 2.9 ตัวอย่างของจุดซึ่งมีการเปลี่ยนแปลงทิศทางตามรหัสแบบลูกโซ่ของฟรีแมน

(ก) จุดที่มีการเปลี่ยนแปลงทิศทางบนเส้นแสดงขอบของอักขระ

(ข) อักขระเดิมของรูปที่ 2.9(ก)

กำหนด $C_i = (C_{x_i}, C_{y_i})$ : $i = 1, \ldots, I$

โดย $C_i$ เป็นลำดับของจุดที่มีการเปลี่ยนทิศทางตามรหัสแบบลูกโซ่ของฟรีแมน

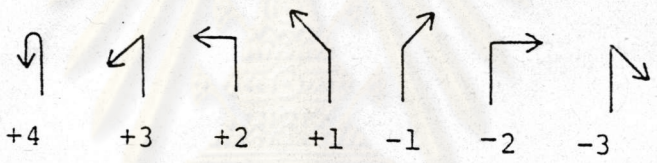
กำหนด $F_i \in \{0, 1, 2, 3, 4, 5, 6, 7\}$

โดย $F_i$ เป็นค่ารหัสแบบลูกโซ่ของฟรีแมน ของจุด $C_i$

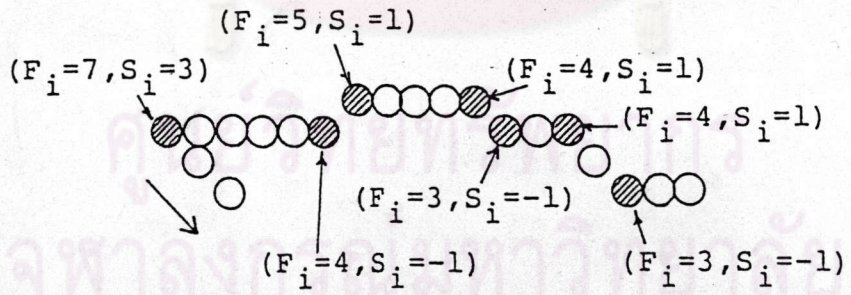
และ $S_i \in \{1, 2, 3, 4, -1, -2, -3\}$

โดย $S_i$ เป็นเครื่องหมายซึ่งจะกำหนดให้กับจุด $C_i$ แต่ละจุด ตามลักษณะการเปลี่ยนทิศทาง ดังแสดงในรูปที่ 2.10

ในรูปที่ 2.11 เป็นตัวอย่างการให้เครื่องหมาย $S_i$ และ ค่ารหัส $F_i$




รูปที่ 2.10 เครื่องหมาย $S_i$




รูปที่ 2.11 ตัวอย่างการให้เครื่องหมาย $S_i$ และ ค่ารหัส $F_i$

การกำหนดจุดบ่งความนูน (vertice +) และจุดบ่งความเว้า (vertice -) ให้แก่จุด $C_i$ (จุดที่มีการเปลี่ยนทิศทาง) เพื่อใช้ในการตัดแบ่งเส้นแสดงขอบของอักขระออกเป็นส่วนโค้งย่อย มีข้อกำหนดดังนี้

ข้อกำหนดที่ 1 : จุดบ่งความนูน จะถูกกำหนดให้แก่จุด $C_i$ ถ้าจุด $C_i$ สอดคล้องกับเงื่อนไขใดเงื่อนไขหนึ่ง ดังต่อไปนี้

1.1 $(S_i > 0)$ และ $(S_{i-1} > 0$ หรือ $S_{i+1} > 0)$

1.2 $(S_i > 0)$ และ $(S_{i-1} < 0)$ และ $(S_{i+1} < 0)$

และ $(F_i \neq FF)$ และ $(DD_{i,i+1} \geq K1)$

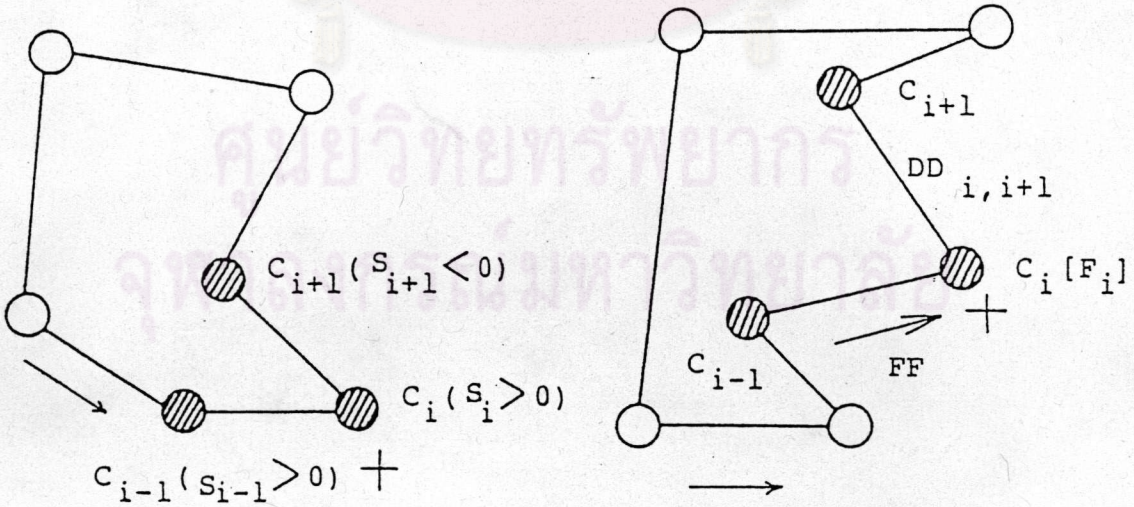
โดย $DD_{i,i+1}$ เป็นระยะห่างระหว่างจุด $C_i$ และ จุด $C_{i+1}$

เมื่อ $DD_{i,i+1}$ = (จำนวนจุดซึ่งอยู่ระหว่างจุดที่ i และ จุดที่ i+1) + 1 (2.1)

และ FF เป็นค่ารหัสแบบลูกโซ่ของฟรีแมนของจุดซึ่งตรวจพบก่อนที่จะถึงจุด $C_i$

K1 เป็นค่าคงที่ ซึ่งได้จากการทดลองกับข้อมูลทดสอบ

ถ้าจุด $C_i$ ถูกกำหนดให้เป็นจุดบ่งความนูน และ $DD_{i,i+1}$ มีค่ามากกว่า K2 แล้ว ค่า FF จะถูกกำหนดให้มีค่าใหม่ตามค่าของ $F_i$ โดย FF จะถูกกำหนดให้มีค่าเริ่มต้นเป็น 4 และ K2 เป็นค่าคงที่ ซึ่งได้จากการทดลองกับข้อมูลทดสอบ ในรูปที่ 2.12 จะเป็นตัวอย่างของจุด $C_i$ ที่ถูกกำหนดให้เป็นจุดบ่งความนูน ตามข้อกำหนดข้างต้น




(ก) (ข)

รูปที่ 2.12 ตัวอย่างการกำหนดจุดบ่งความนูนให้แก่จุด $C_i$

ข้อกำหนดที่ 2 : จุดบ่งความเว้า จะถูกกำหนดให้แก่จุด $C_i$ ถ้าจุด $C_i$ สอดคล้องกับเงื่อนไขใดเงื่อนไขหนึ่ง ดังต่อไปนี้

2.1 $(S_i < 0)$ และ $(S_{i-1} < 0$ หรือ $S_{i+1} < 0)$

2.2 $(S_i < 0)$ และ $(S_{i-1} > 0)$ และ $(S_{i+1} > 0)$

และ $(F_i \neq FF)$ และ $(DD_{i,i+1} \geq K1)$

เมื่อมีการกำหนดจุดบ่งความนูนและจุดบ่งความเว้าให้แก่จุด $C_i$ แล้ว ก็จะทำให้สามารถตัดแบ่งเส้นแสดงขอบของอักขระออกเป็นส่วนโค้งย่อย ตามลักษณะของความเว้าและความนูนได้ต่อไป

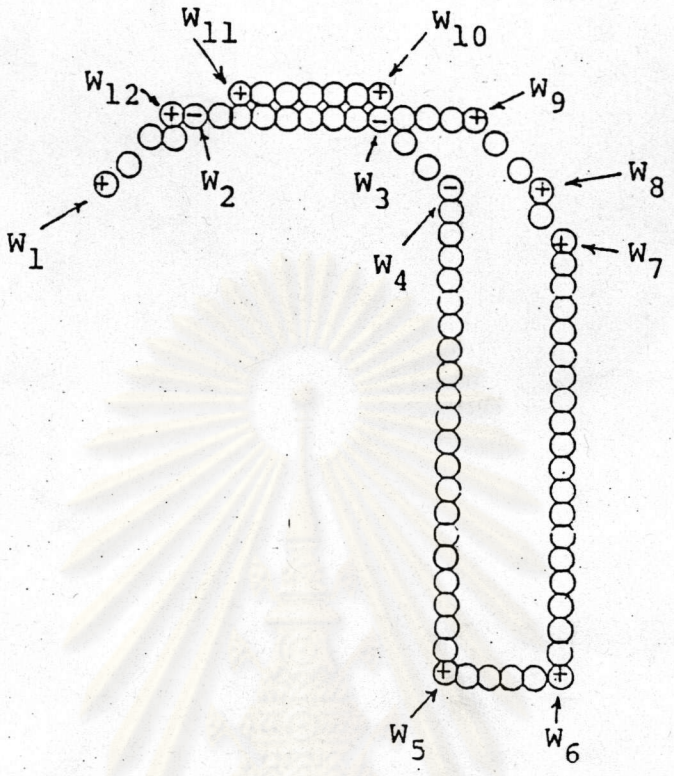
<u>ค่าจำกัดความของส่วนโค้งเว้า และส่วนโค้งนูน</u>

ส่วนโค้งเว้า คือ ส่วนของเส้นโค้งที่ผ่านจุดบ่งความเว้า ซึ่งอยู่ระหว่างจุดบ่งความนูน 2 จุด โดยจุดบ่งความนูนทั้งสองจุดจะเป็นจุดปลายทั้งสองของส่วนโค้งเว้านั้น

ส่วนโค้งนูน คือ ส่วนของเส้นโค้งที่ผ่านจุดบ่งความนูน ซึ่งอยู่ระหว่างจุดบ่งความเว้า 2 จุด โดยจุดบ่งความเว้าทั้งสองจุดจะเป็นจุดปลายทั้งสองของส่วนโค้งนูนนั้น

ตัวอย่างของการกำหนดจุดบ่งความนูน และ จุดบ่งความเว้า แก่อักขระในรูปที่ 2.9(ข) แสดงได้ดังในรูปที่ 2.13

ดังนั้นจุดเริ่มต้น และ จุดสุดท้าย ของส่วนโค้งนูน (หรือ ส่วนโค้งเว้า) จะเป็นจุดบ่งความเว้า (หรือจุดบ่งความนูน) ดังแสดงในรูปที่ 2.13 ซึ่งจะเห็นว่าส่วนที่ผ่าน จุด $W_4$, $W_5$, $W_6$, $W_7$, $W_8$, $W_9$, $W_{10}$, $W_{11}$, $W_{12}$, $W_1$ และ $W_2$ จะเป็นส่วนโค้งนูน และ ส่วนที่ผ่านจุด $W_1$, $W_2$, $W_3$, $W_4$ และ $W_5$ จะเป็นส่วนโค้งเว้า




รูปที่ 2.13 ตัวอย่างของอักขระไทยที่มีการกำหนดจุดบ่งความนูนและจุดบ่งความเว้า

2.3.2 ฟีเชอร์ของอักขระ

เนื่องจากอักขระไทยมีจำนวนมาก ดังนั้นลักษณะทางโกลบอลของเส้นแสดงขอบของอักขระได้แก่ จำนวนหัวของอักขระ (H) จำนวนส่วนของอักขระในกรณีที่อักขระนั้นเขียนในลักษณะแยกจากกัน (R) เช่น ญ จะมีค่า R = 2 จำนวนส่วนโค้งทั้งหมดของอักขระ (Z) คือ ทั้งส่วนโค้งเว้า และส่วนโค้งนูนรวมกัน จะถูกนำมาใช้ในการจัดแบ่งกลุ่มของอักขระ

นอกจากนี้แล้ว วิธีที่นับได้ว่ามีประสิทธิภาพวิธีหนึ่งในการรู้จำอักขระไทย ก็คือ วิธีการในการนำเอาส่วนโค้งย่อย ที่ได้จากการตัดแบ่งเส้นแสดงขอบของอักขระ มาใช้เป็นลักษณะสำคัญในการรู้จำอักขระ (Hiranvanichakorn et al., 1984, 1985) คือ จะมีการตัดแบ่งส่วนโค้งแต่ละส่วนโค้งออกเป็นส่วนย่อยอีกครั้งหนึ่ง โดยจะใช้จุดที่ส่วนโค้งมีการเปลี่ยนทิศทางเป็นตำแหน่งในการตัดแบ่ง ซึ่งจะทำให้ได้ลักษณะทางโลคอลของส่วนโค้งออกมาเช่น ความยาวของแต่ละส่วนที่ตัดแบ่งออกมาได้ ลักษณะทางโลคอลที่ได้ออกมานี้จะถูกนำไปใช้ประโยชน์ในการเปรียบเทียบเพื่อหาส่วนโค้งคู่ที่มีความคล้ายกันมากที่สุด จากนั้นก็จะใช้ส่วนโค้งคู่ที่มีความคล้ายกันมากที่สุดนี้เป็นตำแหน่งเริ่มต้นสำหรับการเปรียบเทียบเพื่อหาอักขระที่มีความคล้ายกันมากที่สุดต่อไป

## 2.4 ไดนามิกโปรแกรมมิง

### 2.4.1 การเปรียบเทียบแบบไดนามิกโปรแกรมมิง

การเปรียบเทียบแบบไดนามิกโปรแกรมมิงถือว่า เป็นอัลกอริทึมในการเปรียบเทียบรูปแบบที่มีการปรับตัวยืดหยุ่นทางแกนเวลา ซึ่งจะทำให้เกิดประสิทธิภาพในการนำไปใช้งาน ดังนั้นไดนามิกโปรแกรมมิงจึงได้มีการนำไปใช้กันอย่างกว้างขวางสำหรับการรู้จำรูปแบบของเสียงพูด และ การรู้จำอักขระแบบออนไลน์ (Sakoe, H. and Chiba, S., 1978) ในการวิจัยนี้จึงได้นำอัลกอริทึมของการเปรียบเทียบแบบไดนามิกโปรแกรมมิงมาประยุกต์ใช้กับการรู้จำอักขระไทยในระบบออฟไลน์ เนื่องจากเมื่อนำมาใช้กับการรู้จำอักขระในระบบออฟไลน์แล้วก็สามารถที่จะทำให้เกิดการปรับตัวยืดหยุ่น ในเรื่องขนาดของตัวพิมพ์อักขระซึ่งมีทั้งขนาดเล็กและขนาดใหญ่ได้ ทำให้สามารถรู้จำอักขระที่มีรูปแบบเดียวกันได้ทั้งขนาดเล็กและขนาดใหญ่ ซึ่งจะเป็นการช่วยลดขั้นตอนของการปรับอักขระให้มีขนาดตามที่กำหนดลงได้ด้วย

### 2.4.2 หลักการของการเปรียบเทียบแบบไดนามิกโปรแกรมมิง

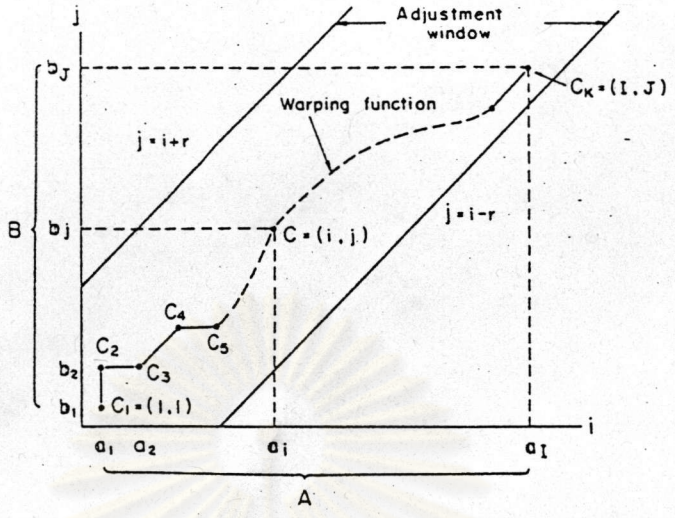
#### 2.4.2.1 คำจำกัดความของความแตกต่างที่มีการปรับตัวยืดหยุ่นทางแกนเวลา

รูปแบบของเสียงพูด สามารถแทนได้โดยการดึงลักษณะสำคัญของรูปแบบนั้นๆ ออกมา และ เขียนแทนในลักษณะที่เป็นลำดับของเวกเตอร์ของรูปแบบ (feature vectors) ในแกนของเวลา ได้ดังนี้

$$A = a_1, a_2, a_3, \ldots, a_i, \ldots, a_I \qquad (2.4ก)$$

$$B = b_1, b_2, b_3, \ldots, b_j, \ldots, b_J \qquad (2.4ข)$$

การแก้ปัญหาในเรื่องความแตกต่างของเวลา (timing difference) ระหว่างรูปแบบ 2 รูปแบบนี้ สามารถอธิบายได้ดังรูปที่ 2.14 บนระนาบ i-j โดยให้แกน i แทนรูปแบบ A และ แกน j แทน รูปแบบ B ในกรณีที่รูปแบบอยู่ในกลุ่มเดียวกัน ความแตกต่างของเวลาระหว่างรูปแบบสามารถแทนได้โดยใช้ลำดับของจุด c = (i,j) ดังนี้




รูปที่ 2.14 ฟังก์ชั่นแวร์ปปิ้ง F และ ข้อจำกัดของการปรับช่วงขอบเขต r

$$F = c(1), c(2), \ldots, c(k), \ldots, c(K) \quad (2.5)$$

โดย $c(k) = (i(k), j(k))$

ลำดับของจุดที่กล่าวถึงนี้ จะใช้แทน ฟังก์ชั่นซึ่งมีการจับคู่ (mapping) จากแกนเวลาของรูปแบบ A ไปบนแกนเวลาของรูปแบบ B และเรียกฟังก์ชั่นในลักษณะนี้ว่า ฟังก์ชั่นแวร์ปปิ้ง (Warping function) ในกรณีที่ไม่มีความแตกต่างของเวลาระหว่างรูปแบบทั้งสอง ฟังก์ชั่นแวร์ปปิ้งก็จะอยู่ในตำแหน่งของเส้นทะแยงมุม $i = j$ และฟังก์ชั่นแวร์ปปิ้งนี้จะเปลี่ยนแปลงไปจากตำแหน่งของเส้นทะแยงมุมถ้าหากมีความแตกต่างของเวลาเกิดขึ้น

ความแตกต่างของฟีเชอร์เวกเตอร์ของรูปแบบ 2 เวกเตอร์ คือ $a_i$ และ $b_j$ คำนวณได้ดังนี้

$$d(c) = d(i,j) = | a_i - b_j | \quad (2.6)$$

ดังนั้นจะได้ ผลรวมของความแตกต่างที่ถูกถ่วงน้ำหนักบนฟังก์ชั่นแวร์ปปิ้ง F ดังนี้

$$E(F) = \sum_{k=1}^{K} d(c(k)) \cdot w(k) \quad (2.7)$$

เมื่อ $w(k)$ เป็นค่าสัมประสิทธิ์ที่ใช้ในการถ่วงน้ำหนัก และมีค่าเป็นบวกเสมอ ซึ่งค่า $w(k)$ นี้จะเป็นค่าที่ทำให้มีการปรับตัวยืดหยุ่นกับค่าของ E(F)

นอกจากนี้แล้ว การคำนวณค่า E(F) ยังเป็นการกระทำเพื่อให้เกิดความเหมาะสมต่อฟังก์ชั่นแวร์ปปิ้ง F ด้วย ซึ่งการคำนวณค่านี้จะให้ค่าที่น้อยที่สุดเมื่อสามารถหาค่าฟังก์ชั่นแวร์ปปิ้ง F ที่แน่นอนได้ เพื่อทำให้ได้ผลลัพธ์ที่ดีที่สุดต่อการปรับตัวของความแตกต่างของเวลา ดังนั้นค่าความแตกต่างระหว่างรูปแบบ A และรูปแบบ B ก็คือค่าความแตกต่างที่น้อยที่สุดที่เหลืออยู่หลังจากกำจัดความแตกต่างของเวลาออกไปแล้ว ซึ่งค่าความแตกต่างที่น้อยที่สุดนี้จะเป็นค่าที่คงที่เสมอโดยไม่ขึ้นกับการปรับตัวยืดหยุ่นทางแกนเวลา นั่นคือความแตกต่างที่มีการปรับตัวยืดหยุ่นทางแกนเวลาระหว่างรูปแบบ A และรูปแบบ B สามารถหาได้ดังนี้

$$D(A,B) = \min_{F} \frac{\sum_{k=1}^{K} d(c(k)) \cdot w(k)}{\sum_{k=1}^{K} w(k)} \qquad (2.8)$$

โดย $\sum_{k=1}^{K} w(k)$ จะถูกใช้เพื่อการถ่วงน้ำหนักสำหรับผลของค่า K

เมื่อ K เป็นจำนวนจุดบนฟังก์ชั่นแวร์ปปิ้ง F

สมการที่ 2.8 นี้นอกจากจะแสดงให้เห็นถึงค่าจำกัดความพื้นฐานของความแตกต่างที่มีการปรับตัวยืดหยุ่นทางแกนเวลาแล้ว คุณลักษณะอันมีประสิทธิภาพที่ได้จากสมการนี้ยังขึ้นกับ การกำหนดฟังก์ชั่นแวร์ปปิ้ง และ ค่าสัมประสิทธิ์ที่ใช้ในการถ่วงน้ำหนักด้วย การวัดค่าความแตกต่างที่มีการปรับตัวยืดหยุ่นทางแกนเวลาของลักษณะสำคัญจะเปลี่ยนแปลงไปตามคุณสมบัติของรูปแบบ ดังนั้นเพื่อหลีกเลี่ยงปัญหาที่อาจจะเกิดขึ้นจึงได้มีการกำหนดเงื่อนไขขึ้น 2 ข้อดังนี้

ก. การสุ่มตัวอย่างของรูปแบบจะสุ่ม ณ ช่วงเวลาที่แน่นอนอันเดียวกัน สำหรับทุกรูปแบบ

ข. เนื่องจากไม่ทราบว่าส่วนใดเป็นส่วนที่แสดงถึงลักษณะสำคัญของรูปแบบ ดังนั้นในกรณีนี้จึงควรจะให้ถือว่าทุกส่วนของรูปแบบเป็นส่วนที่แสดงถึงลักษณะสำคัญของรูปแบบเหมือนกันหมดทุกส่วน

2.4.2.2 ข้อจำกัดของฟังก์ชั่นแวร์ปปิ้ง

ฟังก์ชั่นแวร์ปปิ้ง F ที่กำหนดดังสมการที่ 2.5 นี้ เป็นแบบอันหนึ่งของการปรับตัวยืดหยุ่นทางแกนเวลาของรูปแบบ ดังนั้นจึงควรมีการกำหนดคุณสมบัติของการปรับตัวยืดหยุ่นทางแกนเวลาที่แท้จริงออกมา หรืออีกนัยหนึ่งฟังก์ชั่นแวร์ปปิ้ง F ซึ่งมีการจับคู่จากแกนเวลาของรูปแบบ A ไปบนแกนเวลาของรูปแบบ B ควรจะต้องสามารถเก็บลักษณะทางโครงสร้างที่สำคัญทางแกนเวลาของรูปแบบ A ไว้ให้ได้ และในทำนองกลับกันฟังก์ชั่นแวร์ปปิ้ง F ซึ่งมีการจับคู่จากแกนเวลาของรูปแบบ B ไปบนแกนเวลาของรูปแบบ A ก็ควรจะต้องสามารถเก็บลักษณะทางโครงสร้างที่สำคัญทางแกนเวลาของรูปแบบ B ไว้ให้ได้ด้วยเช่นเดียวกัน โครงสร้างที่สำคัญทางแกนเวลาของรูปแบบจะต้องมีลักษณะของความต่อเนื่อง (continuity) มีความสัมพันธ์กันในเรื่องของเวลาภายในรูปแบบหนึ่งๆ (monotonicity) เงื่อนไขเหล่านี้สามารถแสดงในรูปของข้อจำกัดของฟังก์ชั่นแวร์ปปิ้ง F หรือ จุด c(k) = (i(k),j(k)) ได้ ดังนี้

ก. มีความสัมพันธ์กันในเรื่องของเวลาภายในรูปแบบหนึ่งๆ

$$i(k-1) \leq i(k) \quad \text{และ}$$

$$j(k-1) \leq j(k)$$

ข. มีลักษณะของความต่อเนื่อง

$$i(k) - i(k-1) \leq 1 \quad \text{และ}$$

$$j(k) - j(k-1) \leq 1$$

ผลจากข้อจำกัดของฟังก์ชั่นแวร์ปปิ้ง F ทั้งสองข้อข้างต้น ทำให้เกิดความสัมพันธ์ระหว่างจุดที่อยู่ติดกัน 2 จุด คือ

$$c(k-1) = \begin{cases} (i(k) , j(k)-1), \\ (i(k)-1 , j(k)-1), \\ \text{หรือ} \quad (i(k)-1 , j(k)) \end{cases}$$

ค. มีขอบเขต (boundary)

$$i(1) = 1, \; j(1) = 1 \quad \text{และ}$$

$$i(K) = I, \; j(K) = J$$

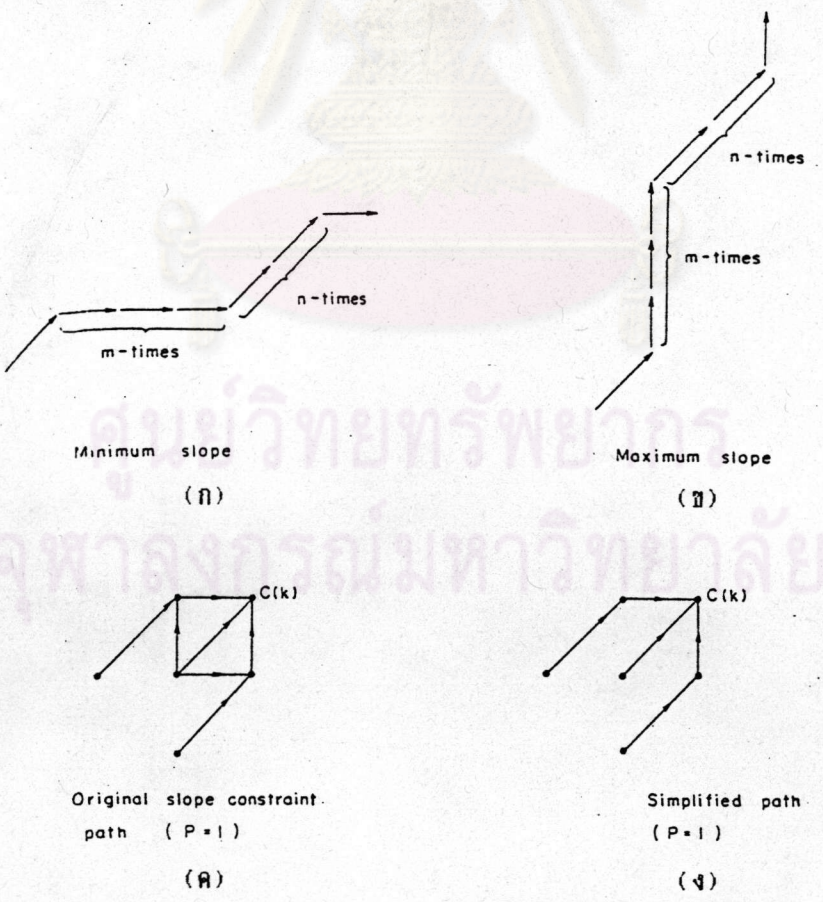
ง. มีการปรับช่วงของขอบเขต r ได้

$$| i(k) - j(k) | \leq r$$

เมื่อ r เป็นจำนวนเต็มบวกที่มีค่าเหมาะสม และเรียกค่านี้ว่า ช่วงของขอบเขต (window length) ซึ่งข้อจำกัดนี้สอดคล้องกับความจริงที่ว่า การปรับตัวยืดหยุ่นทางแกนเวลาโดยทั่วไปแล้วจะไม่ทำให้เกิดความแตกต่างของเวลามากเกินไป

จ. มีข้อจำกัดของความชัน P (slope constraint)

ค่าของความชันนี้ไม่ว่าจะเปลี่ยนแปลงไปอย่างรวดเร็วหรือค่อยๆเปลี่ยนไปก็ตามก็จะมีผลกระทบต่อฟังก์ชั่นแวร์ปปิ้ง F เช่นเดียวกัน ดังนั้นข้อจำกัดนี้จึงควรกำหนดให้เหมาะสมกับฟังก์ชั่นแวร์ปปิ้ง F และ ข้อจำกัดนี้อาจถูกนิจารณาว่าเป็นข้อจำกัดของความสัมพันธ์ที่เป็นไปได้ทั้งหมดระหว่างจุดที่อยู่ติดกันทุกจุดบนฟังก์ชั่นแวร์ปปิ้ง F ดังแสดงในรูปที่ 2.15(ก) และ รูปที่ 2.15(ข)




รูปที่ 2.15 ข้อจำกัดของความชันบนฟังก์ชั่นแวร์ปปิ้ง F

จากรูปที่ 2.15(ก) และ 2.15(ข) สามารถจะอธิบายได้ดังนี้ ถ้าจุด c(k) เคลื่อนไปข้างหน้าในทิศทางของแกน i (หรือ แกน j) เป็นจำนวน m ครั้ง แล้วจุด c(k) จะต้องไม่เคลื่อนไปในทิศทางเดิม จนกว่าจะเคลื่อนไปในทิศทางของเส้นทะแยงมุมเป็นจำนวน n ครั้งเสียก่อน ประสิทธิภาพของความชันสามารถจะคำนวณได้ดังสมการต่อไปนี้

$$P = n / m \qquad (2.9)$$

ถ้าค่า P ยิ่งมากเท่าใดฟังก์ชั่นแวร์ปปิ้ง F ก็จะยิ่งถูกจำกัดมากขึ้นเท่านั้น เมื่อ P = 0 ก็จะไม่มีข้อจำกัดบนฟังก์ชั่นแวร์ปปิ้ง F แต่ถ้าค่าของ P ไม่มีที่สิ้นสุด ($P = \alpha$ ซึ่งก็คือ $m = 0$) แล้ว ฟังก์ชั่นแวร์ปปิ้ง F ก็จะถูกจำกัดอยู่เฉพาะบนแนวเส้นทะแยงมุม $i = j$ เท่านั้น ดังนั้นการกำหนดค่า P จึงควรกำหนดให้เหมาะสม คือไม่มากหรือน้อยเกินไป

จากรูปที่ 2.15(ค) และ 2.15(ง) จะเป็นแนวทางเดินที่เป็นไปได้ของจุด c(k) โดยมีข้อจำกัดของค่าความชัน P = 1 ในรูปที่ 2.15(ค) เป็นรูปที่อธิบายถึงข้อจำกัดข้างต้นได้อย่างชัดเจน ส่วนในรูปที่ 2.15(ง) จะเป็นรูปที่ไม่ได้อยู่ภายใต้ข้อจำกัดเดียวกันกับรูปที่ 2.15(ค) ซึ่งข้อจำกัดใหม่นี้จะช่วยลดจำนวนเส้นทางที่จะต้องค้นหาให้น้อยลง ดังนั้นในระยะหลังจึงมีการนำมาประยุกต์ใช้เช่นกัน แต่ยกเว้นในกรณีที่ P = 0

2.4.2.3 ค่าสัมประสิทธิ์ที่ใช้ในการถ่วงน้ำหนัก (Weighting coefficient)

จากสมการที่ 2.8 จะเห็นได้ว่า ถ้าตัวหารในสมการนี้คือ

$$N = \sum_{k=1}^{K} w(k) \qquad (2.10)$$

ไม่ขึ้นกับฟังก์ชั่นแวร์ปปิ้ง F แล้ว ก็สามารถจะเขียนสมการใหม่ได้ดังนี้

$$D(A,B) = \frac{1}{N} \min_{F} \left[ \sum_{k=1}^{K} d(c(k)) \cdot w(k) \right] \qquad (2.11)$$

สมการที่ได้ใหม่นี้สามารถจะนำมาใช้ได้อย่างมีประสิทธิภาพ โดยใช้เทคนิคของไดนามิกโปรแกรมมิง คำจำกัดความของค่าสัมประสิทธิ์ที่ใช้ในการถ่วงน้ำหนักสามารถแสดงได้เป็น 2 แบบ ดังนี้

ก. แบบสมมาตร (symmetric form)

$$w(k) = (i(k)-i(k-1)) + (j(k)-j(k-1)) \quad (2.12)$$

ดังนั้น $N = I + J$ (2.13)

เมื่อ I และ J เป็นความยาวของรูปแบบ A และ B ตามลำดับ ดังแสดงในสมการที่ 2.4ก และ 2.4ข

ข. แบบไม่สมมาตร (asymmetric form)

$$w(k) = (i(k)-i(k-1)) \quad (2.14)$$

ดังนั้น $N = I$ (2.15)

หรือ $w(k) = (j(k)-j(k-1))$

ดังนั้น $N = J$

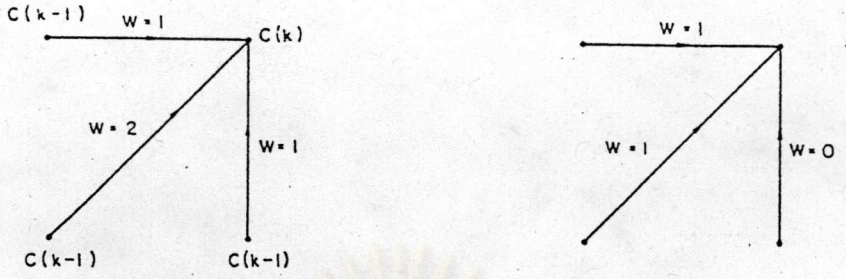
แนวความคิดพื้นฐานของแบบสมมาตรและแบบไม่สมมาตรนี้เริ่มต้นมาจากแนวคิดของ Sakoe และ Chiba (Sakoe et al., 1978)

ถ้าทั้งแกนเวลา i และ j มีลักษณะของความต่อเนื่องแล้ว ในกรณีแบบสมมาตรจะได้ว่า ผลรวมในสมการที่ 2.8 จะหมายถึง ผลรวมของจุดซึ่งอยู่บนแกน $l = i + j$ แต่สำหรับในกรณีแบบไม่สมมาตรแล้วผลรวมดังกล่าว จะหมายถึงผลรวมของจุดซึ่งอยู่บนแกนเวลา i เท่านั้น ผลของความแตกต่างนี้สามารถแสดงให้เห็นในรูปของสมการที่ต่างกันดังนี้

สำหรับแบบสมมาตร : ความแตกต่างที่มีการปรับตัวยืดหยุ่นทางแกนเวลา $D(A,B) = D(B,A)$

สำหรับแบบไม่สมมาตร : ความแตกต่างที่มีการปรับตัวยืดหยุ่นทางแกนเวลา $D(A,B) \neq D(B,A)$

นอกจากนี้แล้วยังมีผลของความแตกต่างที่สำคัญอีกอันหนึ่ง ซึ่งแสดงได้ดังในรูปที่ 2.16




(ก) แบบสมมาตร (ข) แบบไม่สมมาตร

รูปที่ 2.16 ค่าสัมประสิทธิ์ที่ใช้ในการถ่วงน้ำหนัก W(k)
(ก) สำหรับแบบสมมาตร
(ข) สำหรับแบบไม่สมมาตร

จากรูปที่ 2.16(ข) จะเห็นว่าสำหรับแบบไม่สมมาตรนั้น ค่าสัมประสิทธิ์ที่ใช้ในการถ่วงน้ำหนัก w(k) จะลดลงจนกระทั่งมีค่าเป็นศูนย์ เมื่อทางเดินของจุดบนฟังก์ชั่นแวร์ปปิ้งเคลื่อนไปในทิศทางของแกน j หรือ c(k) = c(k-1) + (0,1) แต่ในทางตรงกันข้าม ซึ่งเป็นแบบสมมาตร ดังในรูปที่ 2.16(ก) นั้น จะเห็นได้ว่า ค่าที่น้อยที่สุดของ w(k) จะมีค่าเท่ากับ 1

จากการวิเคราะห์ที่ได้กล่าวมาทั้งหมดนี้ก็เพื่อต้องการจะชี้ให้เห็นว่าแต่ละส่วนของรูปแบบควรจะต้องถูกนำมาใช้ในการพิจารณาอย่างเท่าเทียมกัน และถ้าเป็นไปได้ก็ควรที่จะหลีกเลี่ยงต่อการที่จะไม่นำเอาเวกเตอร์ของรูปแบบใดรูปแบบหนึ่งออกไปจากผลรวมของจุดซึ่งอยู่บนแกนเวลา ดังนั้นถ้าจะสรุปตามผลการวิเคราะห์แล้วก็ควรจะสรุปได้ว่าการใช้วิธีการของไดนามิกโปรแกรมมิงแบบสมมาตรควรจะให้ผลในการรู้จำที่มีความแน่นอน (accuracy) กว่าแบบไม่สมมาตร แต่อย่างไรก็ตามก็ควรจะพิจารณาไว้ด้วยว่าข้อจำกัดของความชันจะมีผลทำให้ความแน่นอนในการรู้จำลดลงเมื่อทางเดินของจุดบนฟังก์ชั่นแวร์ปปิ้งเคลื่อนไปในทิศทางของแกน j ความแตกต่างในเรื่องของประสิทธิภาพระหว่างแบบสมมาตรและแบบไม่สมมาตรนี้จะค่อยๆ หมดไปเมื่อข้อจำกัดของความชันมีค่ามากขึ้น

### 2.4.3 อัลกอริทึมในการทำงานสำหรับการเปรียบเทียบแบบไดนามิกโปรแกรมมิง

#### 2.4.3.1 สมการไดนามิกโปรแกรมมิง

คำจำกัดความของความแตกต่าง ที่มีการปรับตัวยืดหยุ่นทางแกนเวลา D(A,B) ที่ได้ใหม่ดังสมการที่ 2.11 เป็นตัวอย่างของปัญหาหนึ่งซึ่งสามารถนำมาประยุกต์ใช้กับหลักการของการเปรียบเทียบแบบไดนามิกโปรแกรมมิงได้ อัลกอริทึมพื้นฐานสำหรับใช้ในการคำนวณตามสมการที่ 2.11 สามารถเขียนได้ในอีกรูปแบบหนึ่ง ดังนี้

เงื่อนไขเริ่มต้น :

$$g_1(c(1)) = d(c(1)) \cdot w(1) \qquad (2.16)$$

สมการไดนามิกโปรแกรมมิง :

$$g_k(c(k)) = \min_{c(k-1)} [g_{k-1}(c(k-1)) + d(c(k)) \cdot w(k)] \qquad (2.17)$$

ความแตกต่างที่มีการปรับตัวยืดหยุ่นทางแกนเวลา :

$$D(A,B) = \frac{1}{N} g_K(c(K)) \qquad (2.18)$$

จากเงื่อนไขที่กล่าวมานี้ สมมุติให้ $c(0) = (0,0)$
ดังนั้น สำหรับแบบสมมาตร จะได้ว่า $w(1) = 2$
และ สำหรับแบบไม่สมมาตร จะได้ว่า $w(1) = 1$

เมื่อพิจารณาข้อจำกัดของฟังก์ชั่นแวร์ปปิ้งที่ได้กล่าวมา ในหัวข้อที่ 2.4.2.2 และการแทนค่าสัมประสิทธิ์ที่ใช้ในการถ่วงน้ำหนัก $w(k)$ จากสมการที่ 2.12 หรือสมการที่ 2.14 ลงไปในสมการที่ 2.17 แล้ว ก็จะได้อัลกอริทึมมากมายสำหรับใช้ในทางปฏิบัติ

ตัวอย่างที่ง่ายที่สุดตัวอย่างหนึ่ง ได้แก่ อัลกอริทึมแบบสมมาตรที่ไม่มีข้อจำกัดของความชัน (P = 0) แสดงได้ดังนี้

เงื่อนไขเริ่มต้น :

$$g(1,1) = 2d(1,1) \tag{2.19}$$

สมการไดนามิกโปรแกรมมิง :

$$g(i,j) = \min \begin{bmatrix} g(i,j-1) & + d(i,j) \\ g(i-1,j-1) & + 2d(i,j) \\ g(i-1,j) & + d(i,j) \end{bmatrix} \tag{2.20}$$

เงื่อนไขในการปรับช่วงขอบเขต (adjustment window) :

$$j - r \leq i \leq j + r \tag{2.21}$$

ความแตกต่างที่มีการปรับตัวยืดหยุ่นทางแกนเวลา :

$$D(A,B) = \frac{1}{N} g(I,J) \quad ; N = I + J \tag{2.22}$$

2.4.3.2 การคำนวณโดยย่อ

สมการไดนามิกโปรแกรมมิง หรือ g(i,j) จะต้องมีการคำนวณในลักษณะรีเคอร์ซีฟ (recursive) และมีการเรียงของคู่ลำดับ (i,j) จากน้อยไปหามาก โดยเริ่มจาก เงื่อนไขเริ่มต้นที่ (1,1) ไปจนถึง (I,J) สำหรับช่วงที่สมการไดนามิกโปรแกรมมิงจะต้องคำนวณเป็นดังนี้

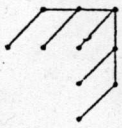
$$1 \leq i \leq I,$$

$$1 \leq j \leq J$$

และ

$$j - r \leq i \leq j + r$$

ตารางที่ 2.1 อัลกอริทึมของสมการไดนามิกโปรแกรมมิงแบบสมมาตรและแบบไม่สมมาตร

| P | Schematic explanation | Symmetric / Asymmetric | DP-equation $g(i,j) =$ | |
|---|---|---|---|---|
| 0 |  | Symmetric | $\min \begin{bmatrix} g(i,j-1)+d(i,j) \\ g(i-1,j-1)+2d(i,j) \\ g(i-1,j)+d(i,j) \end{bmatrix}$ | T.1 |
| | | Asymmetric | $\min \begin{bmatrix} g(i,j-1) \\ g(i-1,j-1)+d(i,j) \\ g(i-1,j)+d(i,j) \end{bmatrix}$ | T.2 |
| 1/2 |  | Symmetric | $\min \begin{bmatrix} g(i-1,j-3)+2d(i,j-2)+d(i,j-1)+d(i,j) \\ g(i-1,j-2)+2d(i,j-1)+d(i,j) \\ g(i-1,j-1)+2d(i,j) \\ g(i-2,j-1)+2d(i-1,j)+d(i,j) \\ g(i-3,j-1)+2d(i-2,j)+d(i-1,j)+d(i,j) \end{bmatrix}$ | T.3 |
| | | Asymmetric | $\min \begin{bmatrix} g(i-1,j-3)+(d(i,j-2)+d(i,j-1)+d(i,j))/3 \\ g(i-1,j-2)+(d(i,j-1)+d(i,j))/2 \\ g(i-1,j-1)+d(i,j) \\ g(i-2,j-1)+d(i-1,j)+d(i,j) \\ g(i-3,j-1)+d(i-2,j)+d(i-1,j)+d(i,j) \end{bmatrix}$ | T.4 |
| 1 |  | Symmetric | $\min \begin{bmatrix} g(i-1,j-2)+2d(i,j-1)+d(i,j) \\ g(i-1,j-1)+2d(i,j) \\ g(i-2,j-1)+2d(i-1,j)+d(i,j) \end{bmatrix}$ | T.5 |
| | | Asymmetric | $\min \begin{bmatrix} g(i-1,j-2)+(d(i,j-1)+d(i,j))/2 \\ g(i-1,j-1)+d(i,j) \\ g(i-2,j-1)+d(i-1,j)+d(i,j) \end{bmatrix}$ | T.6 |
| 2 |  | Symmetric | $\min \begin{bmatrix} g(i-2,j-3)+2d(i-1,j-2)+2d(i,j-1)+d(i,j) \\ g(i-1,j-1)+2d(i,j) \\ g(i-3,j-2)+2d(i-2,j-1)+2d(i-1,j)+d(i,j) \end{bmatrix}$ | T.7 |
| | | Asymmetric | $\min \begin{bmatrix} g(i-2,j-3)+2(d(i-1,j-2)+d(i,j-1)+d(i,j))/3 \\ g(i-1,j-1)+d(i,j) \\ g(i-3,j-2)+d(i-2,j-1)+d(i-1,j)+d(i,j) \end{bmatrix}$ | T.8 |

อัลกอริทึมสำหรับสมการไดนามิกโปรแกรมมิงควรจะต้องมีการปรับปรุงเมื่อนำไปใช้ในแบบไม่สมมาตร หรือ เมื่อมีการนำข้อจำกัดของความชันมาใช้ ในตารางที่ 2.1 จะเป็นการสรุปอัลกอริทึมทั้งสำหรับแบบสมมาตรและแบบไม่สมมาตร เมื่อมีการใช้ข้อจำกัดของความชันที่แตกต่างกัน สมการไดนามิกโปรแกรมมิงสำหรับแบบไม่สมมาตรในตารางนี้จะอยู่ในรูปแบบที่ได้รับการปรับปรุงเรียบร้อยแล้ว

จากตารางที่ 2.1 จะเห็นว่า เมื่อ P = 1 สมการไดนามิกโปรแกรมมิงแบบไม่สมมาตรสมการแรกคือ g(i-1,j-2) + (d(i,j-1)+d(i,j)/2) มีความสอดคล้องกันกับในกรณีของ c(k-1) = (i(k),j(k)-1) และ c(k-2) = (i(k-1)-1,j(k-1)-1) ดังนั้นถ้ายึดตามค่าจำกัดความในสมการที่ 2.14 แล้ว w(k) ก็จะต้องมีค่าเท่ากับศูนย์ ในขณะที่ w(k-1) มีค่าเท่ากับ 1 ซึ่งถือว่าไม่ได้มีการนำค่า d(c(k)) มาใช้ในการคำนวณผลรวมเลย ดังนั้นเพื่อหลีกเลี่ยงเหตุการณ์เช่นนี้ ค่าสัมประสิทธิ์ที่ใช้ในการถ่วงน้ำหนัก w(k-1) = 1 จึงถูกแบ่งออกให้มีค่าอยู่ระหว่างค่าสัมประสิทธิ์ที่ใช้ในการถ่วงน้ำหนัก 2 ตัว คือ w(k-1) และ w(k) ดังนั้น (d(i,j)-1)+d(i,j)) / 2 จึงถูกนำมาใช้แทน d(i,j-1) + 0.d(i,j) ในสมการไดนามิกโปรแกรมมิงแบบไม่สมมาตรสมการแรก และในทำนองเดียวกันก็มีการปรับปรุงในลักษณะที่คล้ายกันนี้กับสมการไดนามิกโปรแกรมมิงแบบไม่สมมาตรสมการอื่นๆ ด้วย ซึ่งการปรับปรุงนี้เป็นสิ่งที่ช่วยทำให้ประสิทธิภาพในการทำงานกับสมการไดนามิกโปรแกรมมิงแบบไม่สมมาตรเพิ่มขึ้นมาก

## 2.5 อักขระไทย

### 2.5.1 อักขระไทยที่ใช้เป็นพื้นฐานสำหรับการรู้จำอักขระ

อักขระไทยประกอบด้วย พยัญชนะ (consonant) 44 ตัว สระ (vowel) 32 รูป วรรณยุกต์ (tonal symbol) 4 รูป และ อักขระพิเศษ (special character) 4 ตัว แต่สระบางรูปสามารถแยกย่อยออก และใช้ในลักษณะอักขระพื้นฐานได้เช่น สระ เ_า เกิดจากสระ เ และ สระ า มาประกอบกัน ส่วนพยัญชนะมียกเว้น 2 ตัว คือ ฃ และ ฅ ซึ่งปัจจุบันไม่มีการนำมาใช้ในชีวิตประจำวันแล้ว ดังนั้นอักขระไทยที่ใช้เป็นพื้นฐานสำหรับการรู้จำจึงเหลือเพียง 67 ตัว คือ พยัญชนะ 42 ตัว สระ 17 ตัว วรรณยุกต์ 4 ตัว และอักขระพิเศษ 4 ตัว ดังนี้

พยัญชนะ 42 ตัว ได้แก่

ก ข ค ฆ ง จ ฉ ช ซ ฌ ญ ฎ ฏ ฐ ฑ ฒ
ณ ด ต ถ ท ธ น บ ป ผ ฝ พ ฟ ภ ม ย
ร ล ว ศ ษ ส ห ฬ อ ฮ

สระ 17 ตัว ได้แก่

ะ า ิ ี ึ ื ุ ู เ แ โ ใ ไ ำ ั ฤ ฦ

วรรณยุกต์ 4 ตัว ได้แก่

่ ้ ๊ ๋

อักขระพิเศษ 4 ตัว ได้แก่

๏ ๎ ฯ ๆ

การรู้จำอักขระไทยนั้นจะมองว่า อักขระไทยส่วนใหญ่ประกอบขึ้นจากเส้นโค้งมีหัวเป็นส่วนประกอบ และ มีอักขระเพียงจำนวนเล็กน้อยเท่านั้นที่เขียนในลักษณะแยกจากกัน เช่น ญ และ ฐ เป็นต้น (Hiranvanichakorn et al., 1985) ดังนั้นในการรู้จำอักขระไทยนั้น ลักษณะสำคัญที่มักจะถูกนำมาใช้ในการวิเคราะห์จึงได้แก่ จำนวนหัวของอักขระ จำนวนส่วนที่เขียนในลักษณะแยกกันของอักขระ และลักษณะของเส้นโค้งที่ประกอบกันเป็นรูปร่างอักขระ

ตารางที่ 2.2 ตารางรหัสอักขระไทยที่ใช้กับคอมพิวเตอร์มาตรฐาน สมอ.

(มาตรฐาน สำนักงานมาตรฐานผลิตภัณฑ์อุตสาหกรรม)

| 4 บิทต้น / 4 บิทท้าย | Ø | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ø | | | SP | Ø | @ | P | | p | | | | ฐ | ภ | ะ | เ | ๐ |
| 1 | | | ! | 1 | A | Q | a | q | | | ก | ฑ | ม | ั | แ | ๑ |
| 2 | | | " | 2 | B | R | b | r | | | ข | ฒ | ย | า | โ | ๒ |
| 3 | | | #F | 3 | C | S | c | s | | | ฃ | ณ | ร | ำ | ใ | ๓ |
| 4 | | | £ | 4 | D | T | d | t | | | ค | ด | ฤ | ิ | ไ | ๔ |
| 5 | | | % | 5 | E | U | e | u | | | ฅ | ต | ล | ี | ๅ | ๕ |
| 6 | | | & | 6 | F | V | f | v | | | ฆ | ถ | ฦ | ึ | ๆ | ๖ |
| 7 | | | ' | 7 | G | W | g | w | | | ง | ท | ว | ื | ็ | ๗ |
| 8 | | | ( | 8 | H | X | h | x | | | จ | ธ | ศ | ุ | ่ | ๘ |
| 9 | | | ) | 9 | I | Y | i | y | | | ฉ | น | ษ | ู | ้ | ๙ |
| A | | | * | : | J | Z | j | z | | | ช | บ | ส | ฺ | ๊ | ฯ |
| B | | | + | ; | K | [ | k | { | | | ซ | ป | ห | | ๋ | ๛ |
| C | | | , | < | L | \ | l | ¦ | | | ฌ | ผ | ฬ | | ์ | |
| D | | | - | = | M | ] | m | } | | | ญ | ฝ | อ | | ํ | |
| E | | | . | > | N | ^ | n | ~ | | | ฎ | พ | ฮ | | ๎ | |
| F | | | / | ? | O | _ | o | | | | ฏ | ฟ | ฯ | ฿ | ๏ | |

2.5.2 กลุ่มอักขระไทยที่มีลักษณะคล้ายคลึงกัน

กลุ่มที่ 1 ได้แก่ ก ถ ภ ฎ ฏ ฤ ฦ

กลุ่มที่ 2 ได้แก่ ค ศ ด ต

กลุ่มที่ 3 ได้แก่ ม ฆ

กลุ่มที่ 4 ได้แก่ บ ป ษ ข ช ซ

กลุ่มที่ 5 ได้แก่ ร ธ

กลุ่มที่ 6 ได้แก่ ณ ฒ ญ

กลุ่มที่ 7 ได้แก่ ฉ น

กลุ่มที่ 8 ได้แก่ พ ฟ ฝ ผ ฝ

กลุ่มที่ 9 ได้แก่ ท ห

กลุ่มที่ 10 ได้แก่ ล ส จ

กลุ่มที่ 11 ได้แก่ อ ฮ ว

กลุ่มที่ 12 ได้แก่ ◌ั ◌ี

กลุ่มที่ 13 ได้แก่ ◌ิ ◌ึ

กลุ่มที่ 14 ได้แก่ ไ ใ โ

2.5.3 <u>กลุ่มอักขระไทยแยกตามระดับการเขียน</u>

ระดับบน ได้แก่

ิ ี ึ ื ่ ้ ๊ ๋ ็ ์ ๎ ํ

ระดับกลาง ได้แก่

ก ข ค ฆ ง จ ฉ ช ซ ฌ ญ ฎ ฏ ฐ ฑ ฒ ณ ด
ต ถ ท ธ น บ ป ผ ฝ พ ฟ ภ ม ย ร ล ว ศ
ษ ส ห ฬ อ ฮ ะ า เ แ โ ใ ไ ๆ ๐ ฤ ๅ

ระดับล่าง ได้แก่

ุ ู